HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Herr

Florian Ehrhardt

Visualisierung von Content Parametern aus biologisch relevanten Texten

Mittweida, 2011

Fakultät

Mathematik, Naturwissenschaften, Informatik

BACHELORARBEIT

Visualisierung von Content Parametern aus biologisch relevanten Texten

Autor:

Herr

Florian Ehrhardt

Studiengang:

Biotechnologie/Bioinformatik

Seminargruppe:

BI08w1

Erstprüfer:

Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:

Daniel Stockmann

Einreichung:

Mittweida, 22.08.2011

Verteidigung/Bewertung:

Mittweida, 2011

Faculty

Mathematics, Natural Sciences, Computer Schiences

BACHELOR THESIS

Visualisization of Content Parameters from biologic relevant Texts

author:

Mr.

Florian Ehrhardt

course of studies:

Biotechnology/Bioinformatics

seminar group:

BI08w1

first examiner:

Prof. Dr. rer. nat. Dirk Labudde

second examiner:

Daniel Stockmann

submission:

Mittweida, 22.08.2011

defence/ evaluation:

Mittweida, 2011

Bibliografische Beschreibung:

Ehrhardt, Florian:

Visualisierung von Content Parametern aus biologisch relevanten Texten. - 2011. - 49 S. Mittweida, Hochschule Mittweida, Fakultät Mathematik, Naturwissenschaften, Informatik, Bachelorarbeit, 2011

Referat:

Die Menge an Daten und den daraus resultierenden Informationen wächst von Jahr zu Jahr enorm an. Daher ist es von großer Bedeutung, Hilfsmittel, Techniken und Tools zu entwickeln, welche die Informationsgewinnung und deren Verarbeitung vereinfachen. Als Grundlage dieser Arbeit dienen biologische Texte. Das Aufspüren von Information und deren Extraktion, ohne den genauen Inhalt der Texte zu kennen, ist ein wichtiger Aspekt in der Informationsverarbeitung. Ebenso werden die Auswirkungen von fehlerhaften Texten auf die Informationsgewinnung betrachtet und ausgewertet.

Danksagung

Der größte Dank geht an meine Fachbetreuer, Prof. Dr. rer. nat. Dirk Labudde und Daniel Stockmann, die mir immer mit Rat und Tat zur Seite standen.

Des Weiteren bedanke ich mich bei meiner Familie, die mir nicht nur während der Bachelorarbeit, sondern auch das gesamte Studium lang den Rücken freigehalten haben. Ohne die familiäre Unterstützung ist so ein Studium nur schwer zu vollenden.

Auch meinen gesamten Kommilitonen möchte ich für diese drei Jahre danken.

Nicht zu vergessen die vielen Menschen, welche in mühevoller Kleinarbeit diese Arbeit gelesen, korrigiert und Verbesserungsvorschläge unterbreitet haben.

Ich wünsche allen Menschen, mit denen ich zusammen diesen Weg gegangen bin, herzlichst alles Gute und viel Glück und Erfolg auf den nächsten Schritten des Lebens.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

IV. Abkürzungsverzeichnis

<CellGE>	<cell gene expression>
<Cge>	<cancer gene expression>
<GEA>	<gene expression analysis>

1 Einleitung

Im Rahmen meines Bachelorstudiums der Biotechnologie/ Bioinformatik an der HS Mittweida beschäftigte ich mich in meiner Bachelorarbeit mit der „Visualisierung von Content Parametern aus biologisch relevanten Texten“. Diese Arbeit baut auf dem Beleg aus dem fünften Fachsemester sowie auf dem Praxismodul, einschließlich der Praxisarbeit, auf und ist somit eine Vertiefung in das Thema „Text Mining“.

Ursprünglich war es angedacht, dieses Thema in Zusammenarbeit mit der Polizeidirektion Chemnitz/Erzgebirge zu bearbeiten. Als Untersuchungsobjekte dieser Arbeit sollten polizeirelevante Texte dienen, die auf ihre Ähnlichkeit hin untersucht werden, um die Arbeit der Polizei bei der Tätersuche zu erleichtern. Leider wurde die Freigabe der polizeilichen Texte durch den Oberstaatsanwalt bis zum heutigen Zeitpunkt nicht gewährt. Daher verwendete ich als Korpus Texte über Genexpression. Ziel war die Extraktion von Wissen aus diesen Texten, das Aufdecken von Zusammenhängen und deren Visualisierung, ohne die Texte gelesen zu haben oder deren genauen Inhalt zu kennen.

Festzuhalten ist, dass die Art der Texte und deren Themenbereiche nicht von Bedeutung für meine Arbeit waren. Es ging um die Techniken und Programme, deren korrekte Arbeitsweise und richtige Anwendung.

2 Grundlagen

2.1 Was ist Text

Zu Beginn steht die Frage: Was ist Text überhaupt? Text repräsentiert Wissen und ist damit die Grundlage der Wissensverarbeitung. Er besteht aus einer Aneinanderreihung von Sätzen, diese aus Wortformen, welche wiederum aus Buchstaben eines definierten Alphabets bestehen. Diese „Zeichenketten“ sind in diesem Zustand nur Daten, erst deren Interpretation nach einem bekannten und genau festgelegten Interpretationsschema machen sie zu Informationen. Wenn man mehrere Informationen miteinander verknüpft und zur Problemlösung einsetzt, dann wird Information als „Wissen“ bezeichnet. [1]

2.2 Was kann man an Texten vergleichen

Um Texte auf ihrer Ähnlichkeit untersuchen zu können, muss man festlegen, was man an ihnen vergleichen will. Eine Möglichkeit dabei ist die Zählung der Übereinstimmungen identischer Wörter. Eine andere Möglichkeit ist, den verwendeten Wortschatz zu überprüfen, also wie viele Fachausdrücke benutzt werden und von welcher Art diese sind. Ebenso kann man die Art und die Anzahl der verwendeten Allgemeinbegriffe und -wörter zählen und dadurch die Ähnlichkeit bewerten. Natürlich gibt es noch mehr Möglichkeiten, auf die ich aber nicht näher eingehen werde, da diese auf meine Arbeit keinen Einfluss haben.

2.3 Begriffsdefinition „Ähnlichkeit“

Der Vergleich von Texten miteinander führt zu der Erkenntnis, dass gewisse Texte mehr Übereinstimmungen haben als andere. Die Anzahl an Übereinstimmungen kann man auch als Ähnlichkeit bezeichnen. Laut Definition bedeutet Ähnlichkeit: In charakteristischen Merkmalen übereinstimmend. [2] Je mehr Merkmale übereinstimmen, in diesem Fall z.B. die Anzahl an identischen Buchstabenfolgen, umso höher ist die Ähnlichkeit.

2.4 Was brauche ich zum Vergleich

Um Texte vergleichen zu können braucht man ein Ähnlichkeitsmaß. Dies ist ein nummerischer Wert, der die Ähnlichkeit zwischen Vektoren angibt. Also muss man die Texte in Vektoren umwandeln. Dabei stellt man aber nicht den ganzen Text als Vektor dar, sondern man vergleicht Worthäufigkeiten in den einzelnen Texten. Diese Häufigkeiten kann man dann als Vektor darstellen. In einem Beispiel werde ich dies verdeutlichen. Tabelle 1 zeigt sechs Texte d1 - d6 und zwei Beispielwörter „cancer“ und „gene", welche in diesen Texten vorkommen. Die Zahlenwerte geben die Häufigkeiten der Wörter in den verschiedenen Texten an.

	d1	d2	d3	d4	d5	d6
cancer	1	1	2	1	1	1
gene	1	1	2	1	5	5

Tabelle 1: Worthäufigkeiten in verschiedenen Texten

Diese Worthäufigkeiten können nun grafisch dargestellt werden:

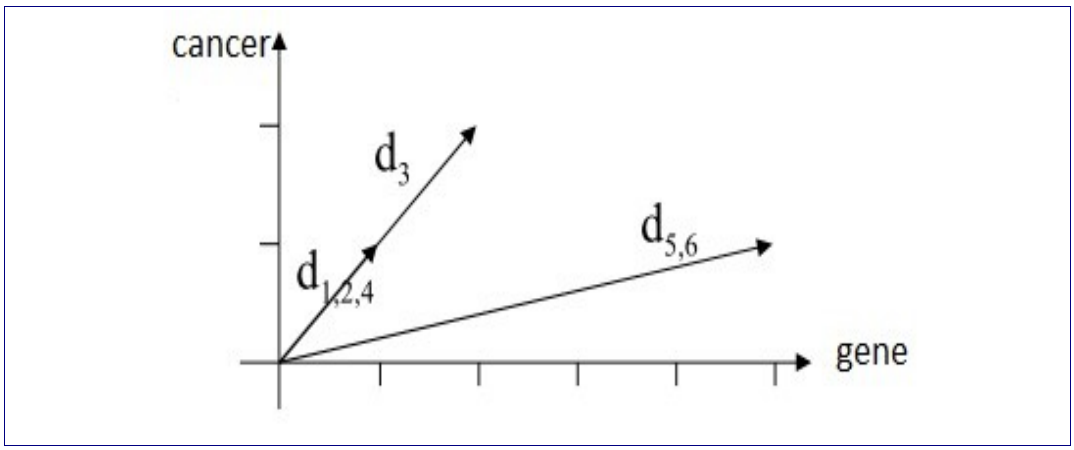


Abbildung 1: Vektordarstellung von Wörtern

Im Text d3 kommen jeweils zweimal die Wörter „cancer“ und „gene“ vor, was die Richtung und die Länge des Vektors angibt. Nun kann man mit verschiedenen Ähnlichkeitsmaßen die Ähnlichkeit bestimmen (Cosinus, Dice, Jaccard usw.). Ich werde nun näher auf das Cosinus-Ähnlichkeitsmaß eingehen, weil dieses für meine Arbeit die besten Ergebnisse lieferte. Anhand der folgenden Formel lässt sich die Similarität zwischen Texten bestimmen:

$$sim(\vec{x},\vec{y}) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{|\vec{x}|\cdot|\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\cdot\sqrt{\sum_{i=1}^{n} y_i^2}} = \vec{x}\cdot\vec{y}$$

Abbildung 2: Cosinus-Ähnlichkeitsmaß

Ausgangspunkt dieser Gleichung ist das Skalarprodukt zwischen zwei Vektoren, zu sehen am Ende der Gleichung. Aus der Formel lässt sich nun ablesen, dass die Similarität zwischen zwei Vektoren genau dem Cosinus-Winkel (cos ($\vec{x}$, $\vec{y}$) zwischen diesen beiden Vektoren entspricht. Der Zähler gibt an, wie gut die beiden Texte korrelieren, das heißt, wie viele Übereinstimmungen sie haben. Im Nenner wird durch die euklidische Länge der Vektoren geteilt, damit man auch unterschiedlich lange Texte miteinander vergleichen kann.
Je größer der Winkel zwischen den Vektoren ist, umso unähnlicher sind sich die Texte. Der Cosinus kann, mathematisch gesehen, Werte zwischen -1 und 1 annehmen, da wir aber nur positive Eingabewerte zulassen, liegt der Ähnlichkeitswert zwischen 0 und 1. [3]

2.5 Gewinnung der Distanzmatrix

Das Programm „Text Similarity 0.08“ ist ein Open Source Tool um die Ähnlichkeit von Texten zu analysieren und ist in PERL geschrieben. Es arbeitet mit dem so genannten TF-IDF-Algorithmus. Im Allgemeinen errechnet der Algorithmus aus der Häufigkeit eines Terms oder Wortes in dem jeweiligen Text und der Häufigkeit des Terms im gesamten Textkorpus ein Maß für die Relevanz des Terms. TF steht dabei für *term frequency* (die Termhäufigkeit in

einem Dokument) und IDF für *inverse document frequency* (die inverse Häufigkeit des Terms im gesamten Textkorpus). Dies geschieht nach folgender Formel:

$$tf\text{-}idf_{t,d} = tf_{t,d} * idf_t$$

Dabei steht das tiefgestellte t für den jeweilig untersuchten Term und das tiefgestellte d für das Dokument. Die sich ergebenden Zahlenwerte kann man nun wieder als Vektor darstellen und die Ähnlichkeit mit dem Cosinus-Ähnlichkeitsmaß berechnen [4].

Das verwendete Programm arbeitet als Grundlage mit diesem Algorithmus, untersucht aber nicht die Häufigkeit (und somit die Relevanz) des Terms über den gesamten Korpus. Sondern es sucht nach so genannten *overlaps* zwischen zwei zu untersuchenden Texten. Es werden also direkte Übereinstimmungen zwischen Wortfolgen in beiden Textdokumenten gesucht. Dabei werden die Textlängen normiert.[5]

Für die Untersuchung werden 30 Texte aus den Bereichen „cell gene expression“ (*CellGE*), „cancer gene expression“ (*Cge*) und „gene expression analysis“ (*GEA*) benutzt. Bei einem nötigen Vergleich aller Texte miteinander entstand eine 30 x 30 Matrix mit 900 Werten. Das Programm errechnete die Ähnlichkeit zwischen Texten, für die weitere Verfahrensweise benötigt man aber die Distanz zwischen den Texten. Nach folgender Formel wurde aus der Ähnlichkeit die Distanz berechnet (ein Auszug aus der entstandenen Matrix befindet sich im Anhang 1):

Distanzwert = 1 – Ähnlichkeitswert

2.6 Erzeugung des Abstandsbaumes mittels PHYLIP

PHYLIP (*PHYL*ogeny *I*nference *P*ackage) ist ein frei verfügbares Programmpaket (geschrieben in der Programmiersprache C), mit dem man die Phylogenie, also die stammesgeschichtliche Entwicklung von Lebewesen, veranschaulichen und somit besser verstehen kann. Es braucht eine genau festgelegte Eingabedatei, welche eine symmetrische Distanzmatrix enthält. Diese wird dann mit der Neighbor-Joining-Methode hierarchisch angeordnet und dargestellt.[6]

Da das Programm PHYLIP Distanzen, also reine Zahlenwerte, darstellt, kann man es auch für den Zweck nutzen, um sich einen Abstandsbaum für Texte generieren zu lassen.

Der aus der Eingabedatei erzeugte Baum ist also die grafische Ausgabe der Ähnlichkeitswerte. Texte mit einer hohen Ähnlichkeit liegen im Baum näher zusammen, als Texte mit einer geringeren Ähnlichkeit. Dabei kommt es vor, dass zwei Texte einen Ast bilden. Diese haben also eine so hohe Ähnlichkeit, dass sie so nah wie möglich zusammen gruppiert werden.

Verwirklicht wird dies mit dem 1987 von Saitou und Nei vorgestellten Neighbor-Joining-Algorithmus. Dieses bottom-up Clusterverfahren geht zunächst von einem sternförmigen Ausgangsbaum aus, bei dem alle Elemente in der Wurzel vereint sind. Nun werden die Elemente, in unserem Fall Distanzen zwischen den einzelnen Texten, mit der geringsten Distanz ausgewählt und zu einem Ast zusammengefasst. Die Distanzmatrix wird neu berechnet und wieder die zwei Texte mit den geringsten Abstand zusammengefasst. Dies geschieht so lange, bis die Sternstruktur vollständig aufgelöst ist und alle Distanzen in den Baum eingefügt wurden.[7] Die Neighbor-Joining-Methode ist im Vergleich zu einem anderem bottom – up Clusterverfahren wie z.B. UPGMA (Unweighted Pair Group Method with Arithmetic mean) wesentlich schneller, die Menge an Eingabedaten, die verarbeitet werden kann, ist wesentlich größer und diese Methode erzeugt einen unbalancierten Baum. Dies ist erforderlich, da die Abstände zwischen den Texten unterschiedlich groß sind und auch in ihren Differenzen sich stark unterscheiden, was zu unterschiedlich langen Astenden führen muss. Bei balancierten Bäumen sind die Astlängen alle gleich, was eine Auswertung hinsichtlich der Textähnlichkeit verhindert.

3 Erzeugung von semantischen Fehlern in Texten

Jeder Text, unabhängig von seinem Thema und Inhalt, wird von einem Menschen verfasst. Dem Menschen können Fehler unterlaufen. Seien es falsche Zeitform oder das einfache Vertauschen von Buchstaben. Daher ist es wichtig, solche Fehler mit einzurechnen und die Richtigkeit der Programme auf fehlerhafte Texte zu kontrollieren. Bei meiner Untersuchung geht es um Rechtschreibfehler, nicht um inhaltlich falsche Dinge. In jeder Sprache gibt es theoretisch eine unendliche Liste an Rechtschreibfehlern. Da ich mit englischsprachigen Texten arbeite, brauche ich allgemein gültige Regeln für Fehler in der Schreibweise englischer Wörter. Aus einer Liste mit solchen Fehlern [8] wurden allgemeine Regeln für Fehler in der englischen Sprache abgeleitet. Eine Liste aller erzeugten Fehler ist in Tab. 2 zu sehen.

Dabei wurden ausschließlich einfache Rechtschreibfehler erzeugt. Dies beinhaltet das Vergessen eines Buchstabens, wie z.B. ein m oder r bei Worten, in denen diese Buchstaben doppelt auftreten würden. Ein Beispiel dafür ist das Wort „occurrence“ (engl. = das Ereignis), welches laut Statistik meist ohne doppeltem r, sondern nur mit einfachem r geschrieben wird. Ebenso wurden Buchstabendreher berücksichtigt. Bei dem Verfassen von Texten am Computer passieren, durch zum Beispiel schnelles Tippen, Buchstabenvertauschungen sehr schnell, daher bildet diese Gruppe den größten Teil der Fehler. Bei Wörtern wie „sieze“ (eigentliche: seize (belegen)), hygeine (eigentliche: hygiene (Hygiene)) kann es zum Beispiel zu einer Buchstabenvertauschung von ie zu ei oder andersherum kommen.

Ursprungsbuchstabenfolge	Änderung
tian	tion
tion	tian
pp	p
rr	r
mm	m
ou	o
z	s
nn	n
ar	er
dg	g
keys	kies
ie	ei
ei	ie
el	le
eable	able
o	oe
sch	sh
tt	t
cracy	cricy
cricy	cracy

Tabelle 2: Erzeugte Fehler

Diese Liste der Vertauschungen ist sehr lang. Fehler in der Schreibweise können auch durch die Aussprache kommen. Manche Wörter werden etwas anders ausgesprochen, als sie geschrieben werden. Ein Beispiel dafür ist „separate“ (engl. = trennen). Bei ungenauer Aussprache des Wortes, könnte man vermuten, dass anstatt dem a ein e vorkommt und das Wort dann wie folgt geschrieben werden würde: „seperate“. Um solche Fehler einzubeziehen, aber nicht alle a in e umzuwandeln, wurde der darauffolgende Buchstabe mit berücksichtigt, in diesem Fall ein r. Nun werden nur die Buchstabenfolgen „ar“ in „er“ umgewandelt.

Eine letzte mögliche Fehlerquelle war der Unterschied zwischen britischen und amerikanischem Englisch. Das Wort „color“ (deutsch = Farbe) wird im

amerikanischen mit einem zusätzlichen u geschrieben, so dass „colour“ entsteht. Wenn also ein Autor eines Textes eine solches Wort verwendet, wird es vom Programm in die britische Form umgewandelt.

Nun steht fest, welche Fehler erzeugt werden sollen. Der nächste Schritt ist die Erarbeitung eines Programmes, welches automatisch diese Fehler in den Text einbaut. Es wurde mittels eines Quellcodes in der Programmiersprache Java realisiert. Zugriff auf das Programm kann über die beigefügte CD erfolgen. Zu Beginn läuft das Programm durch den Text und zählt alle möglichen Fehler. Es werden alle Wörter erfasst, in denen einer der vorher definierten Buchstabenfolgen auftritt und diese Zahl auf der Console ausgegeben. Daraufhin wird der erste Ausdruck, in diesem Fall „tion“, ausgewählt und das Programm läuft nun wieder durch den Text bis es auf das erste Wort mit der Endung „tion“ trifft. Ist das der Fall, erzeugt das Programm eine Zufallszahl. Liegt diese Zufallszahl unterhalb einer vorher festgelegten Fehlerwahrscheinlichkeit, so wird das Wort geändert, liegt sie darüber, bleibt das Wort unverändert. Der Ersatz für „tion“ ist „tian“. Die Fehlerwahrscheinlichkeit wird vorher manuell festgelegte, um eine Abstufung der Texte zu erreichen. Man erzeugt also eine Menge von geänderten Texten mit jeweils unterschiedlichen Fehlerwahrscheinlichkeiten. Je größer diese Wahrscheinlichkeit ist, umso mehr Wörter werden umgewandelt. Ein Beispiel macht diese Thematik begreiflicher:

Das Programm läuft durch den Beispieltext *CellGE1* und errechnet, das theoretisch 46 Fehler möglich sind, d.h. dass 46 Wörter gefunden wurden, in denen mindestens einer der vorher festgelegten Fehler vorkommt. Nun läuft es wieder durch bis es zu dem ersten Wort kommt, welches auf „tion“ endet. In diesem Fall befindet sich das Wort „preparation“ schon in der Überschrift des Textes. Der nächste Schritt ist die Errechnung der Zufallszahl. Ist dies geschehen, wird diese mit der Fehlerwahrscheinlichkeit verglichen und die Endung wird in „tian“ umgewandelt, sollte die Zufallszahl kleiner als die Fehlerwahrscheinlichkeit sein, oder nicht umgewandelt, sollte die Zufallszahl größer als die Fehlerwahrscheinlichkeit sein.

Der zweite Ausdruck ist „tian“. Dieser würde analog zum ersten Schritt zu „tion“ geändert werden. Da das Programm aber im Schritt vorher manche „tion“ in „tian“ umgewandelt hat, würde der Schritt rückgängig gemacht und das ist nicht sinnvoll. Daher bricht die Umwandlung ab, sollte das Programm auf ein „tian“

treffen, welches es im vorhergehenden Schritt noch ein „tion“ war, es also gerade geändert wurde. Nachteilig bei dem Abbruchmechanismus ist, dass das Programm komplett abbricht. Wenn es also auf ein „tian“ trifft, was vorher ein „tion“ war, läuft das Programm nicht weiter durch den Text und sucht andere „tian“, die nicht geändert wurden, sondern macht mit dem nächsten Ausdruck weiter. Somit werden Wörter, in denen regulär der Ausdruck „tian“ vorkommt, und weiter hinten im Text stehen, nicht verändert, was eine Diskrepanz zwischen der Anzahl möglicher Veränderungen und der Anzahl der tatsächlichen Veränderungen verursacht. Auf diese Thematik gehe ich aber später noch einmal ein.

Diese Mechanismen der Umwandlung sowie das Abbrechen bei schon geänderten Wörtern geschehen nun für jeden Ausdruck. Um nicht jeden Text einzeln einlesen und ändern zu müssen, liest das Programm automatisch den kompletten Ordner mit den „Ursprungstexten“ ein, ändert alle darin enthaltenen Texte und schreibt sie in einen neuen Ordner namens „Falsche_Dateien“.

4 Statistische Auswertung der Richtigkeit des Programmes

Um die Genauigkeit des oben beschriebenen Programmes zu überprüfen, wurden zwei Beispieltexte (*CellGE1* und *CGE2*) statistisch ausgewertet. Das Programm zur Fehlererzeugung lief 100 mal für jede Fehlerwahrscheinlichkeit (0,1;0,2...1) durch den Text und erzeugte eine gewisse Anzahl an Fehlern. Dies diente der Untersuchung der erzeugten Fehlerhäufigkeiten für unterschiedliche Fehlerwahrscheinlichkeiten.

4.1 Auswertung für CellGE1

Bei der Auswertung der Fehlerverteilung bei verschiedenen Fehlerwahrscheinlichkeiten fällt auf, dass diese der Normalverteilung sehr ähnlich sehen. Im Text *CellGE1* waren 46 Veränderungen möglich. Bei einer Fehlerwahrscheinlichkeit von 0,2 müssten 9, bei 0,5 müssten 23 und bei 0,8 müssten 37 Fehler eingebaut werden. In den Abbildungen 3, 4 und 5 sind die Fehlerverteilung für diesen Text dargestellt. Abbildung 3 zeigt die Fehlerverteilung für den Text CellGE1 bei einer Fehlerwahrscheinlichkeit von 0,2:

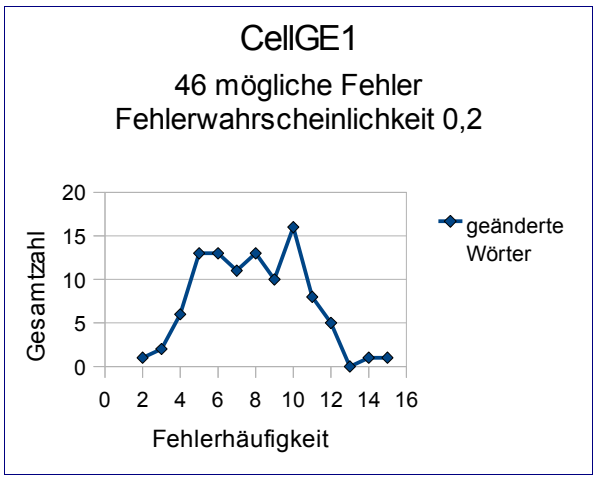


Abbildung 3: Fehlerverteilung für den Text CellGE1 bei einer Fehlerwahrscheinlichkeit von 0,2

Bei einer Fehlerwahrscheinlichkeit von 0,2 müssten rechnerisch etwa 9 Fehler

in den Text eingebaut werden. Im Diagramm ist zu erkennen, dass das Maximum bei 11 Fehlern liegt. Trotzdem wurden in den 100 Durchläufen durchschnittlich 8,83 Fehler eingebaut. Um daraus die tatsächliche Fehlerrate zu berechnen, muss man die durchschnittliche Anzahl der erzeugten Fehler, in diesem Fall 8,83, durch die Anzahl aller möglichen Fehler teilen, in diesem Fall 46. Wenn man dies tut, kommt man auf eine Fehlerrate von 0,19. Dies entspricht fast der Vorgabe von 0,2.

Abbildungen 4 zeigt die Fehlerverteilung des gleichen Textes bei einer Fehlerwahrscheinlichkeit von 0,5:

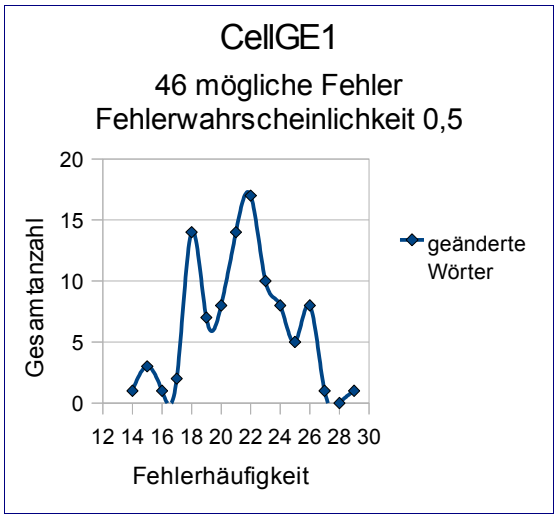


Abbildung 4: Fehlerverteilung für den Text CellGE1 bei einer Fehlerwahrscheinlichkeit von 0,5

Aus dem Diagramm herauszulesen ist, dass die Häufigkeit von 22 Fehlern in 100 Durchläufen am größten ist. Der Fehlerdurchschnitt beträgt 21,33. Das entspricht einer Fehlerrate von 0,464. Diese liegt nur minimal unterhalb der gewollten Fehlerwahrscheinlichkeit von 0,5. Der geringe Unterschied entsteht dadurch, dass zwar 46 Fehler theoretisch möglich sind, in Wahrheit aber nur maximal 39 Fehler erzeugt werden können. Dieser Unterschied wiederum entsteht durch den im vorangegangen Kapitel beschriebenen Sachverhalt, dass geänderte Wörter nicht zurück geändert werden (tion → tian).

Wenn die Fehlerwahrscheinlichkeit weiter angehoben wird, ändert sich die Fehlerverteilung etwas (Abbildung 5):

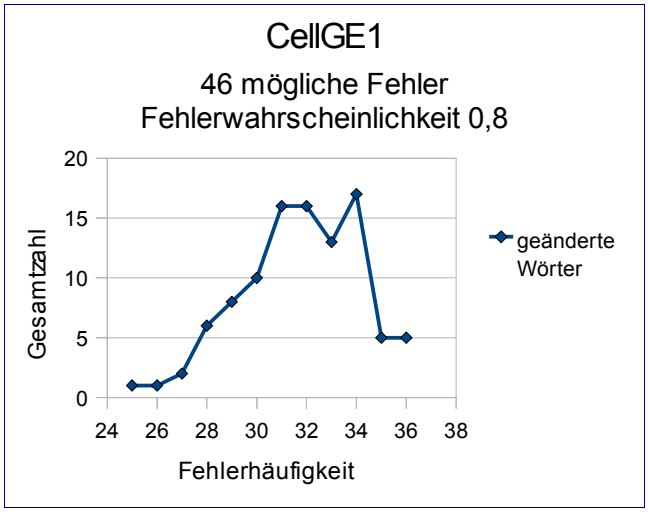


Abbildung 5: Fehlerverteilung für den Text CellGE1 bei einer Fehlerwahrscheinlichkeit von 0,8

Die Grafik ähnelt jetzt nur entfernt einer Normalverteilung. Von den rechnerisch 37 zu erzeugenden Fehlern werden in 100 Durchläufen nur durchschnittlich 31,75 erzeugt, was einer Fehlerrate von 0,69 entspricht. Diese liegt nun doch deutlich unterhalb des gewollten Wertes von 0,8. Auch diese steigende Diskrepanz lässt sich mit dem im Kapitel 3 beschriebenen Sachverhalt erklären. Es wären zwar theoretisch mehr Fehler erzeugbar, die aber nicht alle umsetzbar sind, da sonst schon geänderte Wörter wieder zurück geändert werden würden.

In der folgenden Abbildung 6 sind nun die Unterschiede zwischen Fehlerwahrscheinlichkeit und Fehlerrate für den Text *CellGE1* dargestellt:

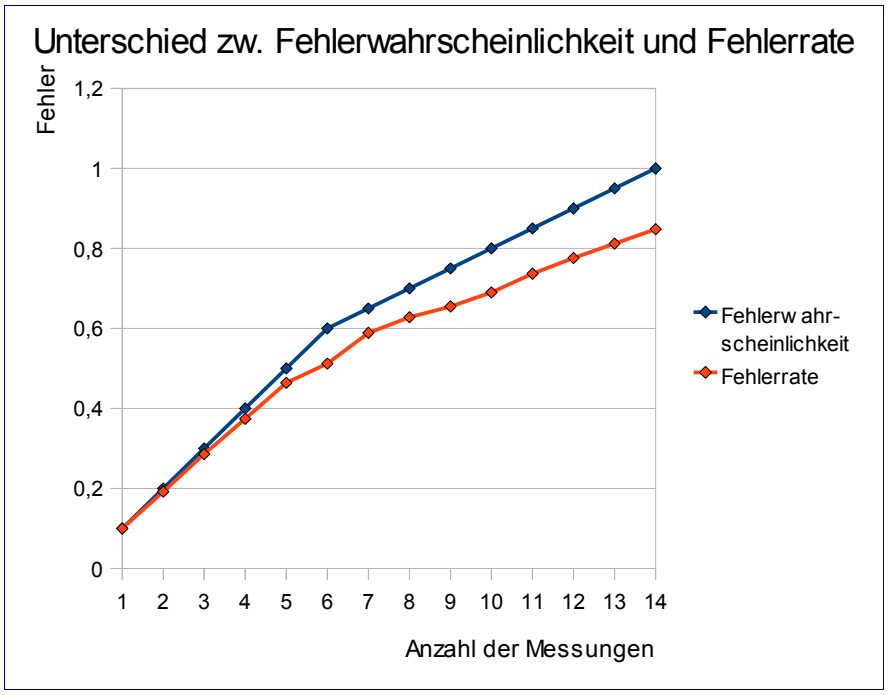


Abbildung 6: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für CellGE1

Die dazu gehörige Tabelle mit den Zahlenwerten befindet sich im Anhang 5.
Anhand dieser Abbildung lässt sich erkennen, dass mit steigender Anzahl der Messungen (Messung = verschiedene Fehlerwahrscheinlichkeiten; 1-6 in 0,1-Schritten (0,1; 0,2..), ab 7 in 0,05-Schritten (0,65; 0,7; 0,75...)) der Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate immer größer wird. Bei einer Fehlerwahrscheinlichkeit von 1 werden 39 Fehler erzeugt, bei 46 Möglichen macht das eine Fehlerrate von 0,848. Wie schon oben beschrieben entsteht diese Diskrepanz durch das Programm.

4.2 Auswertung für den Text CGE2

Die gleiche Untersuchung wie für *CellGE1* wird auch für *CGE2* vollzogen. Jeweils 100 Durchläufe mit unterschiedlichen Fehlerwahrscheinlichkeiten werden unternommen. Exemplarisch werden in den Diagrammen 7, 8 und 9 für die Fehlerwahrscheinlichkeiten 0,2; 0,5 und 0,8 dargestellt.

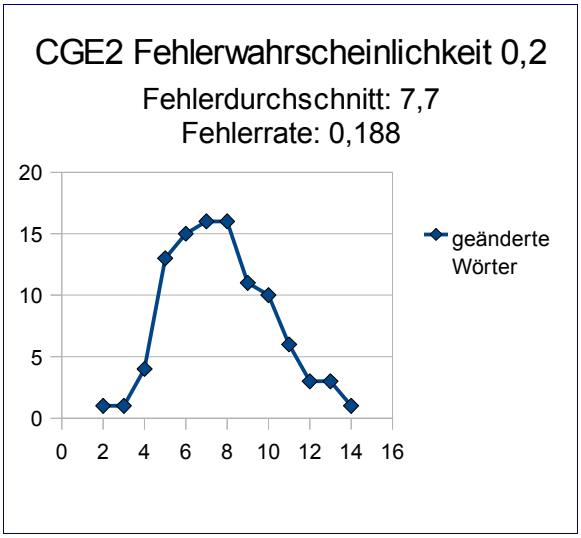


Abbildung 7: Fehlerverteilung für den Text CGE2 bei einer Fehlerwahrscheinlichkeit von 0,2

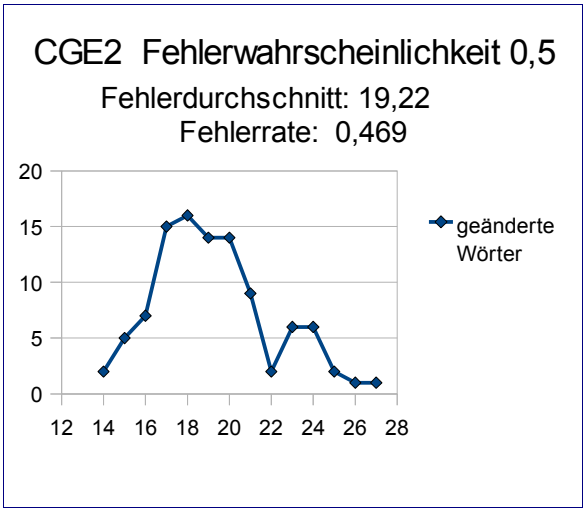


Abbildung 8: Fehlerverteilung für den Text CGE2 bei einer Fehlerwahrscheinlichkeit von 0,5

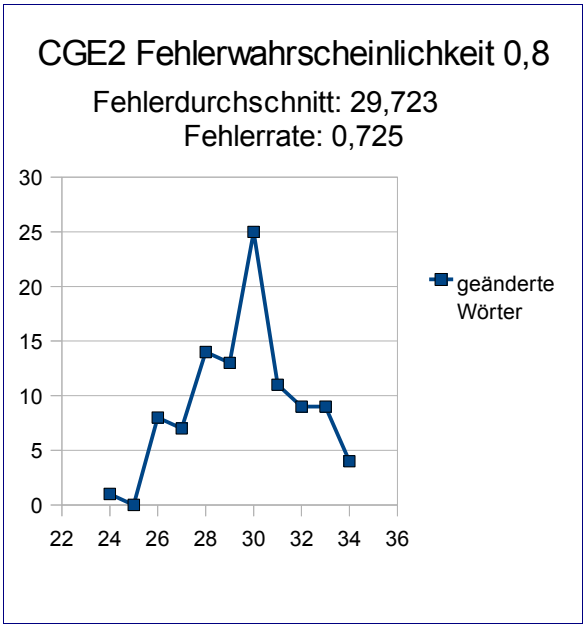


Abbildung 9: Fehlerverteilung für den Text CGE2 bei einer Fehlerwahrscheinlichkeit von 0,8

Die Fehlerrate entsteht wieder durch das Verhältnis von tatsächlich erzeugten und möglichen Fehlern. In diesem Text wären 41 Fehler theoretisch möglich.

Bei einer Fehlerwahrscheinlichkeit von 0,2 erzeugte das Programm durchschnittlich 7,7 Fehler, wobei rechnerisch 8,2 möglich gewesen wären. Dies ergibt eine Fehlerrate von 0,188. Diese weicht nur unwesentlich von den gewollten 0,2 ab.

Durchschnittlich 19,22 Fehler bei rechnerisch 20,5 möglichen ergibt eine Fehlerrate von 0,469 bei einer eingestellten Fehlerwahrscheinlichkeit von 0,5. Auch dieser Wert weicht nur minimal vom Soll ab. Eine etwas größere Abweichung entsteht bei einer Fehlerwahrscheinlichkeit von 0,8. In diesem Fall werden durchschnittlich 29,723 Fehler erzeugt. Dies führt zu einer Fehlerrate von 0,725. Obwohl rechnerisch 32, 8 Fehler entstehen müssten.

Alle diese Abweichungen entstehen durch den eingebauten Stopp im Programm, wenn dieses auf ein Wort trifft, welches es in einem der vorhergehenden Schritte schon geändert wurde.

Im folgenden Diagramm 10 wird nun wieder der Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate dargestellt. Wie schon für den Text *CellGE1* bezeichnet die „Anzahl der Messungen“ die einzelnen Fehlerwahrscheinlichkeiten in den entsprechenden Abstufungen. Auch hier nimmt die Diskrepanz zwischen Fehlerwahrscheinlichkeit und Fehlerrate mit steigender Fehlerwahrscheinlichkeit zu. Der Grund dafür ist auch dafür wieder der eingebaute Stopp des Programmes. Die zugehörige Tabelle mit den Zahlenwerten befindet sich im Anhang 6.

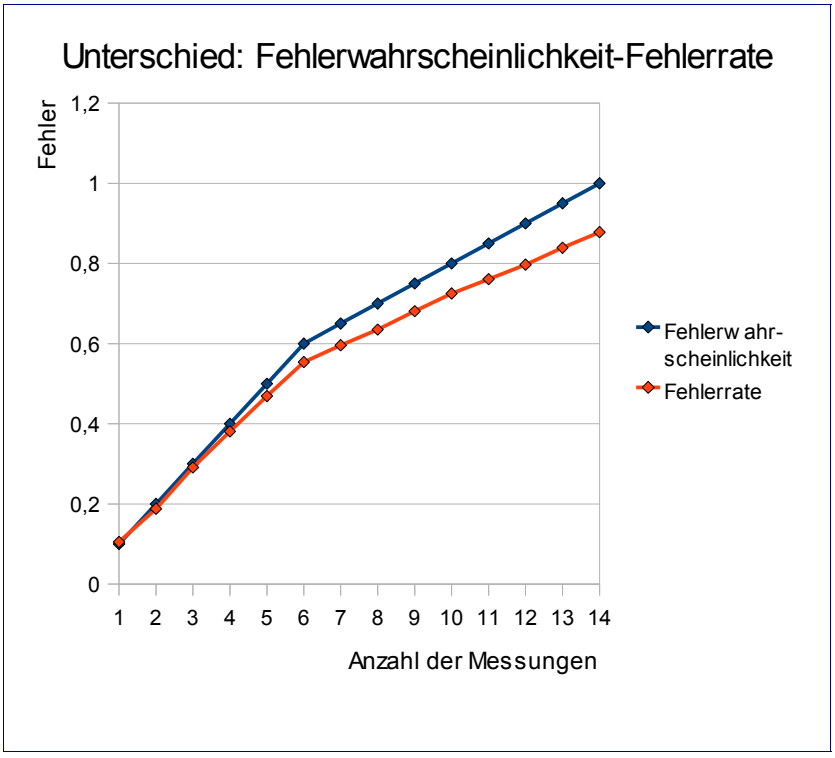


Abbildung 10: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für den Text CGE2

Um zu überprüfen, ob dieses Phänomen der Diskrepanz zwischen Fehlerwahrscheinlichkeit und Fehlerrate allgemein gültig ist oder nur bei diesen beiden Texten, wurde für alle 30 Texte der Unterschied zwischen beiden ermittelt. Als Fehlerwahrscheinlichkeit wurde 1 angegeben. Im Anhang 7 sind sowohl die entstehende Tabelle als auch das Diagramm zur Verteilung von möglichen und tatsächlichen Fehler dargestellt.

Es ist zu erkennen, dass es Texte mit einer sehr hohen Fehlerrate gibt, z.B. bei dem Text *Cge8* liegt diese bei 0,980. Das bedeutet, dass fast alle möglichen

Fehler geändert wurden. Dieser Text ist also fast frei von dem oben beschriebenen Problem des „zurück tauschens“. Andererseits gibt es auch Texte mit einer geringen Fehlerrate, wie z.B. *GEA18* mit 0,776. Bei diesem Text gibt es also viele Wörter, die vorher als Fehler gezählt wurden, dann aber nicht geändert werden konnten, weil sie schon umgeändert wurden.

Bei einer Fehlerwahrscheinlichkeit von 1 baut das Programm durchschnittlich 0,913 Fehler in die Texte ein. Dies ist ein hoher Wert und spricht für die Wirksamkeit und Richtigkeit des Programmes. Der beschriebene Fehler lässt sich nicht vermeiden, da sonst die im ersten Schritt geänderten Wörter im zweiten Schritt wieder in ihre Ausgangsform gebracht werden, was dazu führen würde, dass diese am Anfang zwar gezählt aber nicht geändert werden. Das führt zu einer geringeren Fehlerrate, da insgesamt weniger Wörter geändert werden.

5 Ergebnisse

5.1 Ausgabe von Phylip

Das Programm Phylip erzeugt aus der verwendeten Distanzmatrix einen Abstandsbaum. Dafür wird das enthaltene Paket „neighbor“ verwendet. Dieses erzeugt nach der Ausführung zwei Dateien, „outtree“ und „outfile“. In „outfile“ wird der Baum dargestellt, so wie man ihn im Anhang 2 sehen kann. „Outtree“ erzeugt die Klammernotation, in der die Textdateien und die jeweiligen Abstände aufgeführt sind.

Das Ergebnis der ersten Arbeitsschritte ist ein wohl strukturierter Abstandsbaum, welcher im Anhang 2 und 3 zu sehen ist. Der Baum hat keine Wurzel, da keiner der Texte als Ursprung in Frage kommt, von dem sich alle anderen Texte ableiten lassen (unrooted tree). Das Ende der jeweiligen Äste bilden die so genannten Blätter, also die Texte mit der geringsten Distanz zueinander. Bei der Ausgabe des Baumes als Textdatei (Anhang 2) befinden sich zwischen den Astenden Zahlen, die angeben, in welcher Reihenfolge die Äste in den Baum eingefügt wurde. An dem obersten Ast steht eine 1, dies bedeutet, dass die Texte *CellGE2* (cell gene expression) und *GEA2* (gene expression analysis) die geringste Distanz haben und somit als erstes Paar in den Baum eingefügt wurden. In diesem Fall ist die Distanz 0, da beide Texte, obwohl mit unterschiedlichen Suchbegriffen gefunden, identisch sind. So könnte man jetzt jedes einzelne Astpaar aus dem Baum nehmen und die errechneten Zahlenwerte mit dem Eintrag in den Baum vergleichen und man würde feststellen, dass diese aufsteigend nach ihren Distanzwerten eingeordnet worden sind. Bei der Darstellung mittels njPlot (Anhang 3) ist nicht zu erkennen, in welcher Reihenfolge die Äste eingefügt wurden.

An einem Beispiel wird die Richtigkeit der Ausgabe überprüft: die Texte *Cge5* (cancer gene expression) und *GEA17* bilden keinen gemeinsamen Ast und haben eine Distanz von 0,800623. Wenn man nun beide Texte auf gemeinsame Wörter und Wortgruppen hin untersucht, fällt auf, dass von den je 313 Wörtern pro Text jeweils etwa 65 rot markiert wurden. Die rote Markierung gibt übereinstimmende Worte an. Das sind etwa 20%. Daraus folgt eine

Ähnlichkeit von 20% was eine Distanz von 0,8 bedingt. Da dieser Umstand auch bei anderen Texten festzustellen ist, lässt sich sagen, dass die Berechnung der Distanzen mit dem verwendeten Programm der Richtigkeit entspricht. Zur Überprüfung befinden sich die beiden Texte mit den jeweiligen Markierungen im Anhang 4.
Aus dem Baum kann man nun herauslesen, welche Textgruppe mit welcher anderen besonders eng und häufig verbunden ist. Dabei fällt auf, dass häufig an einem Ast die Texte *CellGE* und *GEA* angeordnet sind. Abgesehen davon sind natürlich die Texte einer Textgruppen eng miteinander verbunden. Eine Ausnahme nehmen dabei die Cge-Texte ein. Diese sind im gesamten Baum und mit vielen anderen Texten verbunden. Dies lässt die Vermutung zu, dass viele Genexpressionsversuche mit Krebsgenen gemacht werden, was eine Verbindung zu vielen Themenbereichen der Genexpression zur Folge hat.

5.2 Auswertung der Bäume von Texten mit erzeugten Fehlern

Eines der Hauptziele dieser Arbeit ist es, die Auswirkungen von erzeugten Fehlern auf die Texte und deren Ähnlichkeiten zu untersuchen. Die 30 Texte werden jeweils mittels des oben beschriebenen Programmes verändert, wobei verschiedene Fehlerwahrscheinlichkeiten benutzt werden. Wenn man nun aus den Texten mittels Phylip Bäume erzeugt, kann man diese mit dem Ausgangsbaum, also dem Baum, in dem die Texte keine erzeugten Fehler haben, vergleichen. Der Ausgangsbaum ist in Anhang 3 dargestellt.
In meiner Auswertung beziehe ich mich auf Bäume mit Texten, die mit einer Fehlerwahrscheinlichkeit von 0,2; 0,5 und 0,8 bearbeitet werden. Diese sind im Anhang 8, 9 und 10 dargestellt.
Wenn man nun diese Bäume miteinander vergleicht, stellt man fest, dass sich rein optisch gesehen nichts verändert. Es bilden sich keine neuen Gruppen oder Abspaltungen. Betrachtet man die einzelnen Strukturen des Baumes genauer, fällt auf, dass die Texte, die einen Ast im Ausgangsbaum bilden, auch in den veränderten Bäumen einen Ast bilden. Die beiden ähnlichsten Texte, *GEA2* und *CellGE2,* bilden immer einen Ast, egal mit welcher

Fehlerwahrscheinlichkeit gearbeitet wurde. Gleich verhält es sich beispielsweise mit den Texten *GEA14* und *Cge7* im unteren Teil des Baumes. Diese bleiben immer zusammen. Die einzige Änderung zum Ausgangsbaum ist, dass sich die Stellung der Paare im Baum verändert. Im Anhang 11 und 12 ist dies exemplarisch dargestellt, wobei Anhang 11 den Ausgangsbaum und Anhang 12 den Baum mit der Fehlerwahrscheinlichkeit 0,5 zeigt. Im Ausgangsbaum ist der Ast mit *Cge1* und *CellGE7* unterhalb der Gruppe mit dem Text *Cge6* und dem Ast *Cge2* und *CellGE8* angeordnet, im Baum mit der Fehlerwahrscheinlichkeit 0,5 darüber. Texte, die keinen Ast bilden, sondern so in den Baum eingefügt wurden, verändern ihre Position nicht. Dies gilt für alle Texte am oberen und unteren Rand des Baumes, sowie in der Mitte.
Im Ergebnis ist zu erkennen, dass die Texte die zu einem Ast zusammen gruppiert wurden vom selben Autor stammen. Die zwei ähnlichsten Texte, *CellGE2* und *GEA2* wurden beide von Jun KR, Lee JN, Park JA, Kim HR, Shin JH, Oh SH, Lee JY, Song SA verfasst. Ein anderes Beispiel für diesen Sachverhalt sind die Texte *CGE9* und *GEA15*. Diese wurden von Lu C. und Cheng SY. Verfasst.

5.3 Auswertung der Bäume von Texten mit unterschiedlichen Fehlerwahrscheinlichkeiten

Eine der wichtigsten Untersuchungen dieser Arbeit ist die Erstellung und Auswertung von Bäumen, deren Texte aus den drei verschiedenen Gruppen *CellGE*, *CGE* und *GEA* stammen, aber jeweils andere Fehlerwahrscheinlichkeiten haben. Im Anhang 13 ist so ein Baum dargestellt. Für diesen Baum wurden aus allen drei Textgruppen jeweils 10 Texte ausgewählt, jeweils mit den Fehlerwahrscheinlichkeiten von 0,1 bis 1. Deutlich zu erkennen ist, dass die jeweiligen Textgruppen als Hauptgruppen zusammen bleiben, die Fehlerwahrscheinlichkeiten führen also nicht zu einer Vermischung der Texte untereinander. Ebenso fällt auf, dass die Textgruppen *CellGE* und *CGE* näher zusammen gruppiert sind.
Für den Baum im Anhang 14 werden wieder jeweils 10 Texte aus den drei Gruppen ausgewählt. Diesmal aber mit einer zufälligen Fehlerwahrschein-

lichkeit. Es werden im Gegensatz zu den obenstehenden Betrachtungen nicht nur „runde“ Wahrscheinlichkeiten (0,1; 0,2; 0,5...) betrachtet, sondern auch differenziertere Wahrscheinlichkeiten, wie 0,03; 0,66 oder 0,73. Auch bei diesem Baum fällt auf, dass die Gruppen zusammen bleiben. Wieder entstehen keine optischen Veränderungen oder Vertauschungen im Baum.

Nimmt man nun nicht aus jeder Gruppe gleich viele Texte, ändert sich der Baum, wie im Anhang 15 zu sehen ist. Dieser Baum entsteht aus drei Texten der Gruppe *CellGE*, 14 Texte von *GEA* und 13 Texten von *CGE* mit jeweils zufällig ausgewählten Fehlerwahrscheinlichkeiten. Auch hier erkennt man deutlich die Gruppenzusammengehörigkeit. Die Textgruppen bleiben im Großen und Ganzen zusammen. Ganz unten im Baum gruppieren sich wieder die schon im Ausgangsbaum einen Ast bildeten Texte *GEA14* und *CGE7*.

6 Auswertung und Diskussion

Als erster wichtiger Punkt und als eine Hauptvoraussetzung für diese Arbeit muss man festhalten, dass keiner der Texte gelesen wurde. Der jeweils spezielle Inhalt ist nicht bekannt, alle Ergebnisse beruhen auf dem Auswertungen der Programme und Tools.

Die Vertauschung von Gruppen im Baum entsteht durch die veränderten Distanzwerte der Texte, welche durch die Fehlererzeugung entstehen. Dadurch werden die einzelnen Texte in einer leicht veränderten Reihenfolge in den Baum eingefügt, was zu einer veränderten Anordnung führt, welche aber auf die Informationsgewinnung keinen Einfluss nimmt.

Die Texte, die ein gemeinsames Astende im Ausgangsbaum bilden, bleiben auch nach der Fehlererzeugung als Astende zusammen. Der Grund dafür ist, dass sie von den gleichen Autoren verfasst wurden. Daraus lässt sich schließen, dass jeder Autor oder jedes Autorenkollektiv seinen eigenen Schreibstil hat. Sei es die Wortwahl, der Satzbau oder die Thematik des Textes. Man könnte also, wenn man mehrere Texte von den gleichen Autoren besitzt, diese aus einem großen Textkorpus mittels dieser Untersuchung heraus filtern. Dies ist ein wichtiges Werkzeug für die Erstellung von Übersichten über die Veröffentlichungen eines Autors/ Autorenkollektivs.

Bei der Auswertung eines Baumes mit jeweils 10 Texten aus allen drei Gruppen mit jeweils gleicher Fehlerwahrscheinlichkeitsverteilung ist zu erkennen, dass keine Vermischung oder Abspaltung von Texten stattfindet. Die Schlussfolgerung ist, dass sich die Erzeugung von Fehlern in einem Text nur gering auf die Ähnlichkeit zu anderen Texten auswirkt. Wenn alle Texte mit der gleichen Fehlerwahrscheinlichkeit verändert werden, entstehen in allen Texten etwa gleich viele Fehler, was dazu führt, dass keine markante und gravierende Veränderung oder Verschiebung im Baum stattfindet. Die Ähnlichkeit von den Texten bleibt bestehen und ist aus dem Baum zu erkennen.

Die nahe Gruppierung der Textgruppen *CellGE* und *CGE* im Baum zueinander, auch mit geänderten Texten, unterstützt die im Kapitel 5.1 aufgestellte These, dass viele Genexpressionsversuche mit Krebsgenen gemacht werden. Texte

der beiden Textgruppen sind sich also inhaltlich so nah, dass auch eine Erzeugung von Fehlern in diesen Texten die Kernaussage nicht verändert und die Ähnlichkeit zueinander bestehen bleibt.

Das gleiche Phänomen tritt auch bei der Untersuchung und Auswertung der Bäume von Texten mit differenzierteren Wahrscheinlichkeiten auf. Auch hier finden keine gravierenden Veränderungen im Baum statt. Es macht also in diesem Fall keinen Unterschied, mit welchen Dezimalschritten die Texte verändert wurden.

Bis hierhin ist festzustellen, dass wenn man jeweils 10 Texte aus jeder Textgruppe nimmt, diese als thematische Gruppe zusammenhängen und dabei die Fehlerwahrscheinlichkeiten und deren Verteilung keine Rolle spielen. Die Texte sind als zusammenhängende thematische Einheit zu erkennen.

Bei der Untersuchung mit jeweils verschiedener Anzahl an Texten bleibt die thematische Gruppenzusammengehörigkeit bestehen. Auch die Verbindung von Texten, die schon im Ausgangsbaum einen Ast bildeten, in diesem Fall *CGE7* und *GEA14*, bleibt bestehen. Dies lässt sich wiederum auf das Autorenkollektiv zurückführen. Diese Übereinstimmungen sind so „stark“, dass sich ihre Texte auch mit verschiedenen Fehlerraten wieder zusammen gruppieren und einen Ast bilden.

7 Zusammenfassung

Mit meinen Untersuchungen konnte ich zeigen, dass die Extraktion von Informationen aus Texten, ohne deren genauen Inhalt zu kennen, möglich ist. Ebenso ist es möglich, Verbindungen und Zusammenhänge zwischen verschiedenen Texten aufzudecken und zu interpretieren. Auch die Auswirkungen von zufällig erzeugten, semantischen Fehlern auf die Textähnlichkeit wurden untersucht. Dabei stellte sich heraus, dass auch fehlerhafte Texte ihre ursprünglichen Verbindungen und Ähnlichkeiten beibehalten und Wissen extrahiert werden kann. Diese Information ist für das eigentliche Anwendungsgebiet der polizeilich relevanten Texte entscheidend. Es ist also möglich, aus fehlerhafte Texte bzw. Texten in einer „künstlichen“ Sprache wie z.B. der Jugendsprache, Informationen zu gewinnen und auszuwerten.

Aus einem erzeugten Baum, der aus jeweils 10 Texten aus jeder Textgruppe mit gleichen oder zufälligen Fehlerwahrscheinlichkeiten besteht, kann man schlussfolgern, dass die jeweiligen 10 Texte einer Textgruppe eine in sich geschlossene Gruppe bilden. Es kommt also nicht zu einer Verbindung von fehlerhaften Texten zu einem Ast oder zu einer Vermischung der Texte untereinander. Dies lässt den Schluss zu, dass sich die Texte einer Textgruppe am nächsten sind und auch erzeugte Fehler daran nichts ändern.

Kommen nun nicht aus jeder Textgruppe gleich viele Texte, sondern ist eine Textgruppe nur mit wenigen Texten vertreten, bleibt im Großen und Ganzen diese Situation bestehen. Aber es kommt an manchen Stellen zu Verbindungen von Texten verschiedener Textgruppen zu einem Ast oder ähnlichem. Die erzeugten Fehler in einem Text führen mit der ungleichen Verteilung der Häufigkeiten von Texten einer Textgruppe zu neuen Verbindungen. Dies verdeutlicht, dass die Fehlererzeugung Auswirkungen auf die Interpretation und den Inhalt von Texten hat.

8 Ausblick

Meine Erkenntnisse sind nur der Anfang des sehr umfangreichen Themengebiets des Text Mining. In meiner Bachelorarbeit habe ich gezeigt, dass es möglich ist, aus unbekannten Texten, deren Inhalt einem nicht bekannt ist, Informationen zu gewinnen, diese auszuwerten und darzustellen. Mit Hilfe geeigneter Tools lassen sich Texte unterschiedlicher Herkunft vergleichen und Informationen herausfiltern. Die Informationsgewinnung und deren Extraktion aus Texten, ohne deren speziellen Inhalt zu kennen, ist ein wichtiger Aspekt und vereinfacht den Umgang mit großen Datenmengen und Textkorpora. Die daraus resultierenden weiteren Untersuchungen, vor allem diese für den angedachten Endabnehmer Polizei, sind sehr umfangreich und stellten keinen Anteil an dieser Arbeit dar.

Im weiteren Verlauf der Zusammenarbeit zwischen der HS Mittweida und der Polizeidirektion Chemnitz/Erzgebirge werden meine Erkenntnisse auf dem Gebiet des Text Mining auf polizeiliche Texte angewandt und weiter bearbeitet und verfeinert. Im weiteren Verlauf können nicht nur lange Texte betrachtet werden, sondern auch SMS oder E-Mails von Tatverdächtigen untersucht werden, da es wesentlich wahrscheinlicher ist, dass Verbrecher auf diesem Wege kommunizieren. Am Ende steht das große Ziel, aus polizeilichen Mitschnitten, also aus gesprochener Sprache, Informationen zu gewinnen und so noch schneller potenzielle Verbrecher aufzuspüren oder Straftaten zu verhindern. Die Grundlagen dafür habe ich anhand biologischer Texte in dieser Arbeit gelegt.

9 Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmitteln angefertigt habe.

Mittweida, den 22.08.2011 Florian Ehrhardt

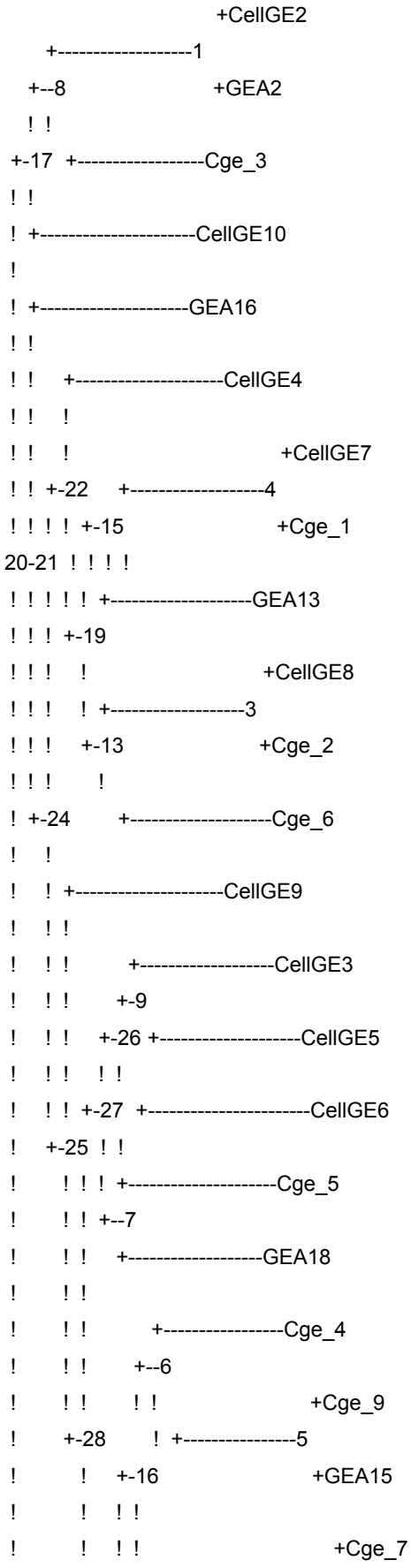
10 Anhang

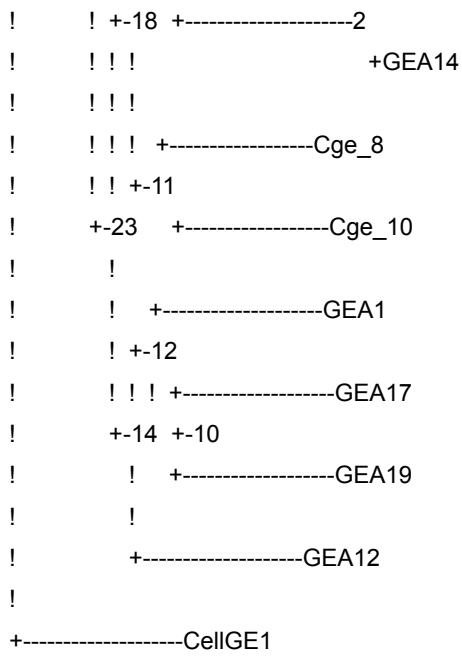
Anhang 1: Auszug aus der Distanzmatrix

Texte	CellGE1	CellGE2	CellGE3	CellGE4	CellGE5
CellGE1	0,000000	0,736220	0,721101	0,743381	0,752151
CellGE2	0,736221	0,000000	0,759874	0,758204	0,760837
CellGE3	0,721101	0,759874	0,000000	0,730519	0,702550
CellGE4	0,743381	0,758204	0,730519	0,000000	0,751534
CellGE5	0,752152	0,760837	0,702550	0,751534	0,000000
CellGE6	0,769634	0,812405	0,747851	0,767081	0,771117
CellGE7	0,746988	0,778157	0,743178	0,708260	0,775417
CellGE8	0,722008	0,752475	0,757387	0,711375	0,740795
CellGE9	0,715499	0,778175	0,741611	0,760148	0,759494
CellGE10	0,725322	0,747292	0,807107	0,746741	0,802233
Cge 1	0,742972	0,774744	0,743178	0,708260	0,772382
Cge 2	0,725869	0,749175	0,754277	0,711375	0,740795
Cge 3	0,719457	0,664151	0,746032	0,727096	0,781095
Cge 4	0,745247	0,775244	0,741935	0,748744	0,767103
Cge 5	0,783582	0,740385	0,763994	0,792422	0,816356
Cge 6	0,697115	0,769841	0,792976	0,737166	0,767764
Cge 7	0,724272	0,767828	0,728125	0,747440	0,751479
Cge 8	0,707317	0,726248	0,696049	0,725166	0,711816
Cge 9	0,693694	0,740602	0,736380	0,704854	0,745455
Cge 10	0,695749	0,730841	0,723776	0,741313	0,736842
GEA1	0,756906	0,752773	0,748503	0,732899	0,792614
GEA2	0,736220	0,000000	0,759874	0,758204	0,760837
GEA12	0,711575	0,756098	0,717791	0,732441	0,750000
GEA13	0,734127	0,753378	0,758347	0,721739	0,789474
GEA14	0,743590	0,774790	0,746835	0,768166	0,769461
GEA15	0,693694	0,744361	0,736380	0,704854	0,745455
GEA16	0,710526	0,757353	0,752151	0,741935	0,773096
GEA17	0,726236	0,742671	0,745008	0,711893	0,772926
GEA18	0,736937	0,766719	0,729412	0,725240	0,751397
GEA19	0,714783	0,752640	0,717143	0,730650	0,782609

Tabelle 3: Auszug aus der Distanzmatrix

Anhang 2: Ausgabe mittels PHYLIP




+-18 +----------2
+GEA14
+----------Cge_8
+-11
+-23 +----------Cge_10
+----------GEA1
+-12
+----------GEA17
+-14 +-10
+----------GEA19
+----------GEA12
+----------CellGE1

Anhang 3: Ausgabe mittels njPlot

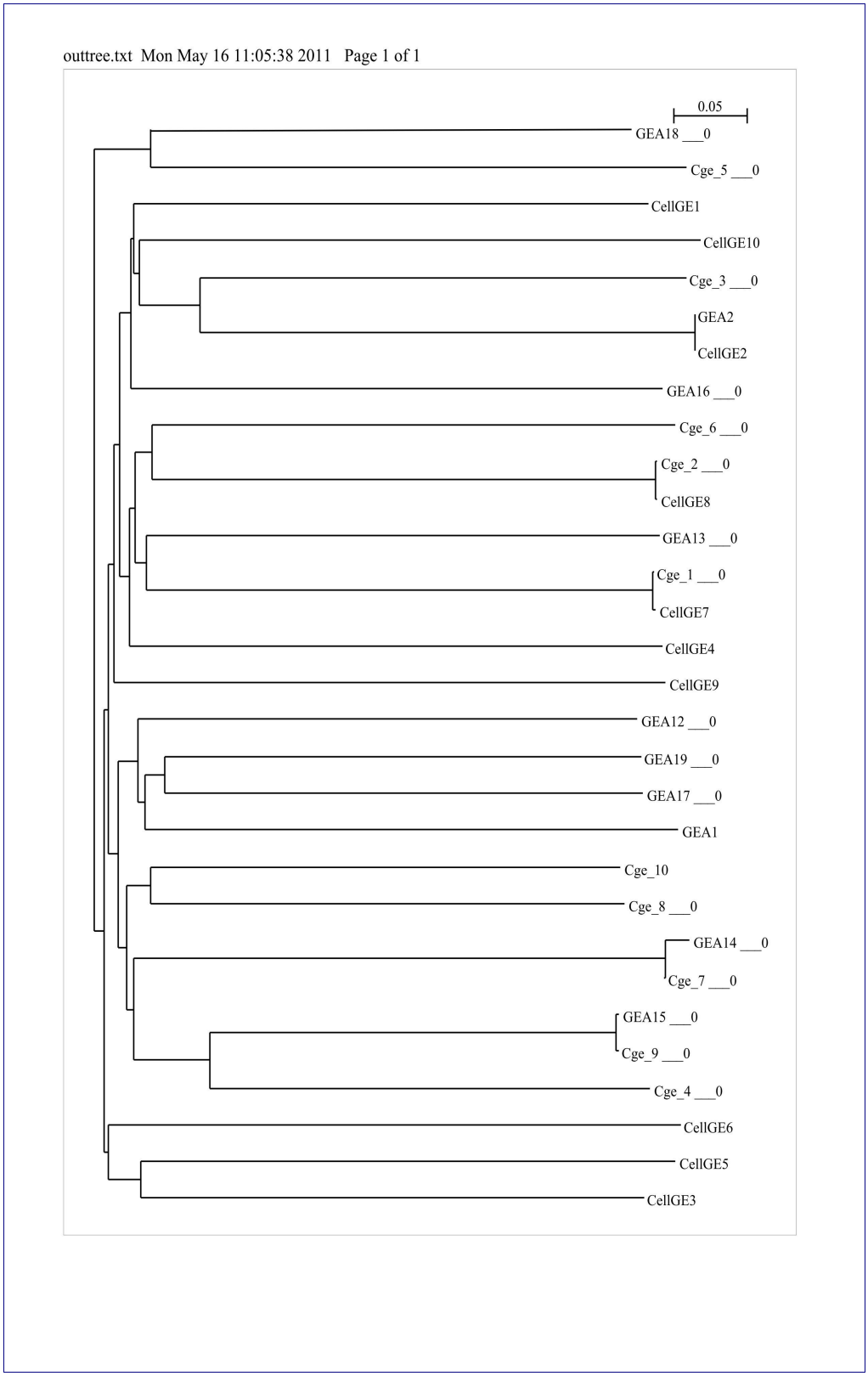
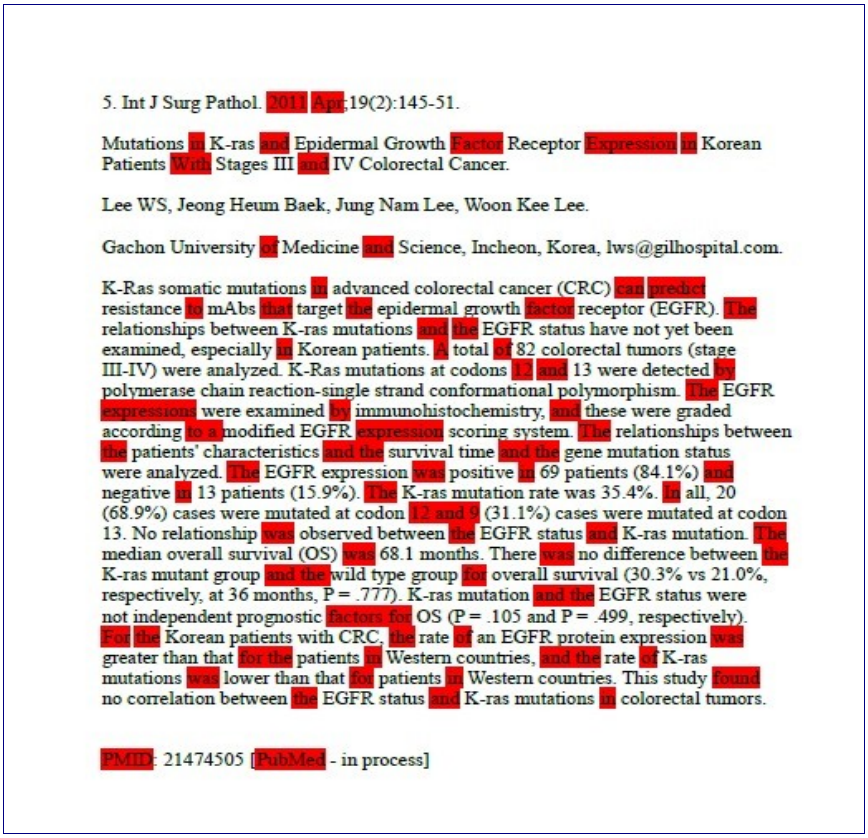


Abbildung 11: Ausgangsbaum

Anhang 4: Beispieltexte mit Markierungen



5. Int J Surg Pathol. 2011 Apr;19(2):145-51.

Mutations in K-ras and Epidermal Growth Factor Receptor Expression in Korean Patients With Stages III and IV Colorectal Cancer.

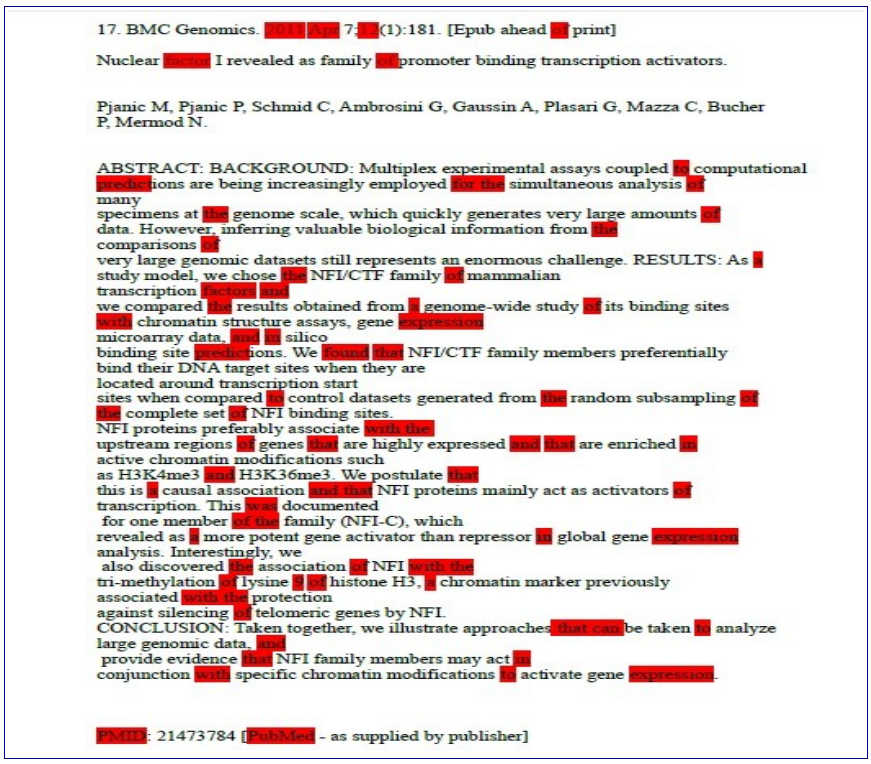
Lee WS, Jeong Heum Baek, Jung Nam Lee, Woon Kee Lee.

Gachon University of Medicine and Science, Incheon, Korea, lws@gilhospital.com.

K-Ras somatic mutations in advanced colorectal cancer (CRC) can predict resistance to mAbs that target the epidermal growth factor receptor (EGFR). The relationships between K-ras mutations and the EGFR status have not yet been examined, especially in Korean patients. A total of 82 colorectal tumors (stage III-IV) were analyzed. K-Ras mutations at codons 12 and 13 were detected by polymerase chain reaction-single strand conformational polymorphism. The EGFR expressions were examined by immunohistochemistry, and these were graded according to a modified EGFR expression scoring system. The relationships between the patients' characteristics and the survival time and the gene mutation status were analyzed. The EGFR expression was positive in 69 patients (84.1%) and negative in 13 patients (15.9%). The K-ras mutation rate was 35.4%. In all, 20 (68.9%) cases were mutated at codon 12 and 9 (31.1%) cases were mutated at codon 13. No relationship was observed between the EGFR status and K-ras mutation. The median overall survival (OS) was 68.1 months. There was no difference between the K-ras mutant group and the wild type group for overall survival (30.3% vs 21.0%, respectively, at 36 months, P = .777). K-ras mutation and the EGFR status were not independent prognostic factors for OS (P = .105 and P = .499, respectively). For the Korean patients with CRC, the rate of an EGFR protein expression was greater than that for the patients in Western countries, and the rate of K-ras mutations was lower than that for patients in Western countries. This study found no correlation between the EGFR status and K-ras mutations in colorectal tumors.

PMID: 21474505 [PubMed - in process]

Abbildung 12: Text CGE5 mit farblichen Markierungen



17. BMC Genomics. 2011 Apr 7;12(1):181. [Epub ahead of print]

Nuclear factor I revealed as family of promoter binding transcription activators.

Pjanic M, Pjanic P, Schmid C, Ambrosini G, Gaussin A, Plasari G, Mazza C, Bucher P, Mermod N.

ABSTRACT: BACKGROUND: Multiplex experimental assays coupled to computational predictions are being increasingly employed for the simultaneous analysis of many specimens at the genome scale, which quickly generates very large amounts of data. However, inferring valuable biological information from the comparisons of very large genomic datasets still represents an enormous challenge. RESULTS: As a study model, we chose the NFI/CTF family of mammalian transcription factors and we compared the results obtained from a genome-wide study of its binding sites with chromatin structure assays, gene expression microarray data, and in silico binding site predictions. We found that NFI/CTF family members preferentially bind their DNA target sites when they are located around transcription start sites when compared to control datasets generated from the random subsampling of the complete set of NFI binding sites. NFI proteins preferably associate with the upstream regions of genes that are highly expressed and that are enriched in active chromatin modifications such as H3K4me3 and H3K36me3. We postulate that this is a causal association and that NFI proteins mainly act as activators of transcription. This was documented for one member of the family (NFI-C), which revealed as a more potent gene activator than repressor in global gene expression analysis. Interestingly, we also discovered the association of NFI with the tri-methylation of lysine 9 of histone H3, a chromatin marker previously associated with the protection against silencing of telomeric genes by NFI. CONCLUSION: Taken together, we illustrate approaches that can be taken to analyze large genomic data, and provide evidence that NFI family members may act in conjunction with specific chromatin modifications to activate gene expression.

PMID: 21473784 [PubMed - as supplied by publisher]

Abbildung 13: Text CGE5 mit farblichen Markierungen

Anhang 5: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für CellGE1

Fehlerwahrscheinlichkeit	Fehlerdurchschnitt	Fehlerrate
0,1	4,6	0,100
0,2	8,83	0,192
0,3	13,14	0,286
0,4	17,22	0,374
0,5	21,33	0,464
0,6	23,54	0,512
0,65	27,11	0,589
0,7	28,89	0,628
0,75	30,15	0,655
0,8	31,75	0,690
0,85	33,89	0,737
0,9	35,7	0,776
0,95	37,35	0,812
1	39	0,848

Tabelle 4: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für CellGE1

Anhang 6: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für CGE2

Fehlerwahrscheinlichkeit	Fehlerrate
0,1	0,105
0,2	0,188
0,3	0,291
0,4	0,381
0,5	0,469
0,6	0,554
0,65	0,596
0,7	0,635
0,75	0,681
0,8	0,725
0,85	0,761
0,9	0,797
0,95	0,839
1	0,878

Tabelle 5: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für CGE2

Anhang 7: Unterschied zwischen tatsächlichen und möglichen Fehlern für alle 30 Texte

Text	mögliche Fehler	tatsächliche Fehler	Fehlerrate
CellGE1	46	39	0,848
CellGE2	37	34	0,919
CellGE3	42	41	0,976
CellGE4	52	47	0,904
CellGE5	71	70	0,986
CellGE6	70	56	0,800
CellGE7	41	35	0,854
CellGE8	41	36	0,878
CellGE9	39	37	0,949
CellGE10	34	33	0,971
cge1	41	35	0,854
cge2	41	36	0,878
cge3	34	30	0,882
cge4	45	43	0,956
cge5	35	32	0,914
cge6	32	31	0,969
cge7	40	38	0,950
cge8	50	49	0,980
cge9	38	37	0,974
cge10	36	29	0,806
GEA1	51	47	0,922
GEA2	37	34	0,919
GEA12	56	45	0,804
GEA13	45	41	0,911
GEA14	38	37	0,974
GEA15	38	37	0,974
GEA16	45	43	0,956
GEA17	48	46	0,958
GEA18	49	38	0,776
GEA19	55	52	0,945

Tabelle 6: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für alle Texte

Diagramm:

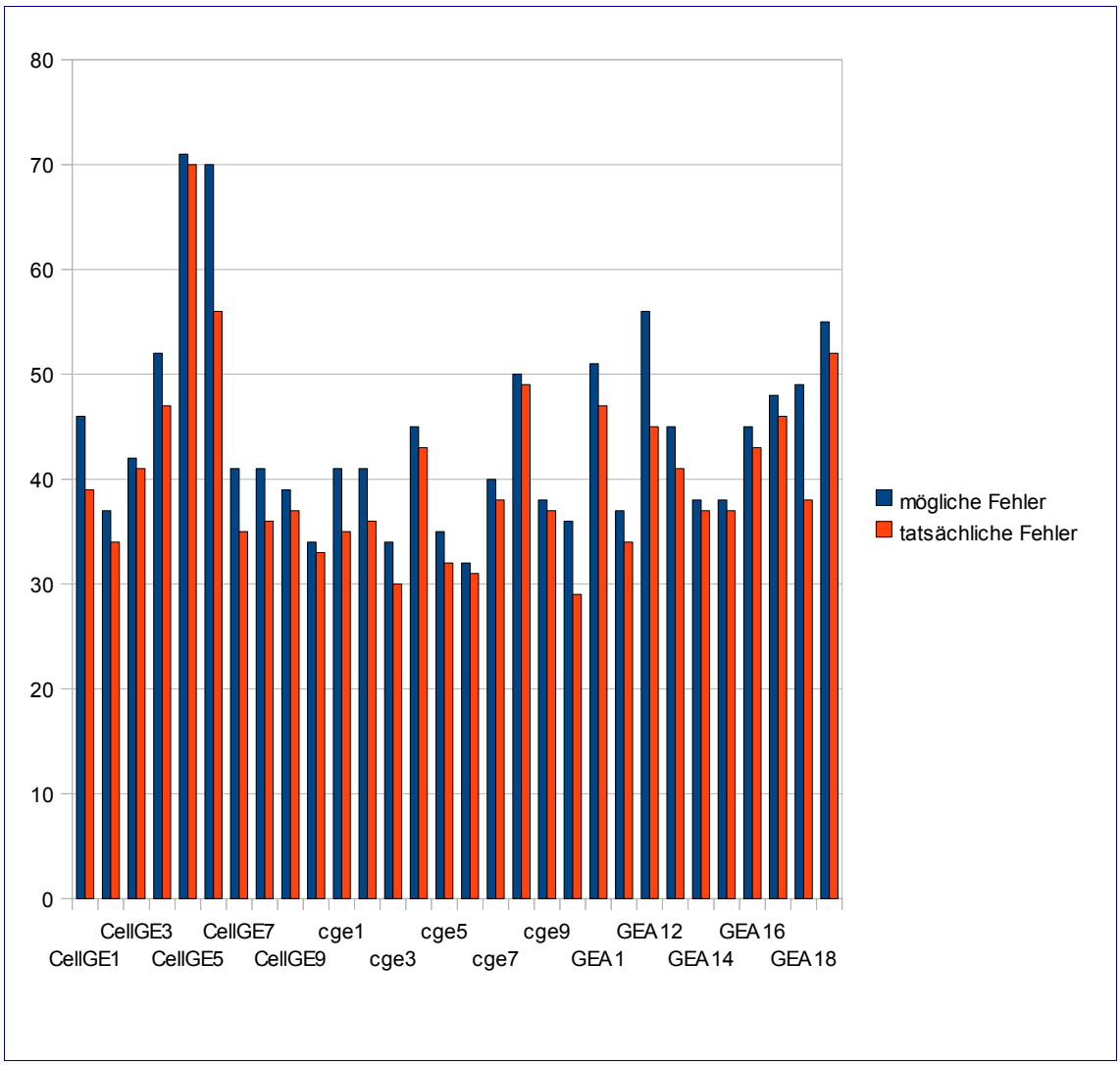


Abbildung 14: Unterschied zwischen Fehlerwahrscheinlichkeit und Fehlerrate für alle Texte

Anhang 8: Baum mit Fehlerwahrscheinlichkeit 0,2

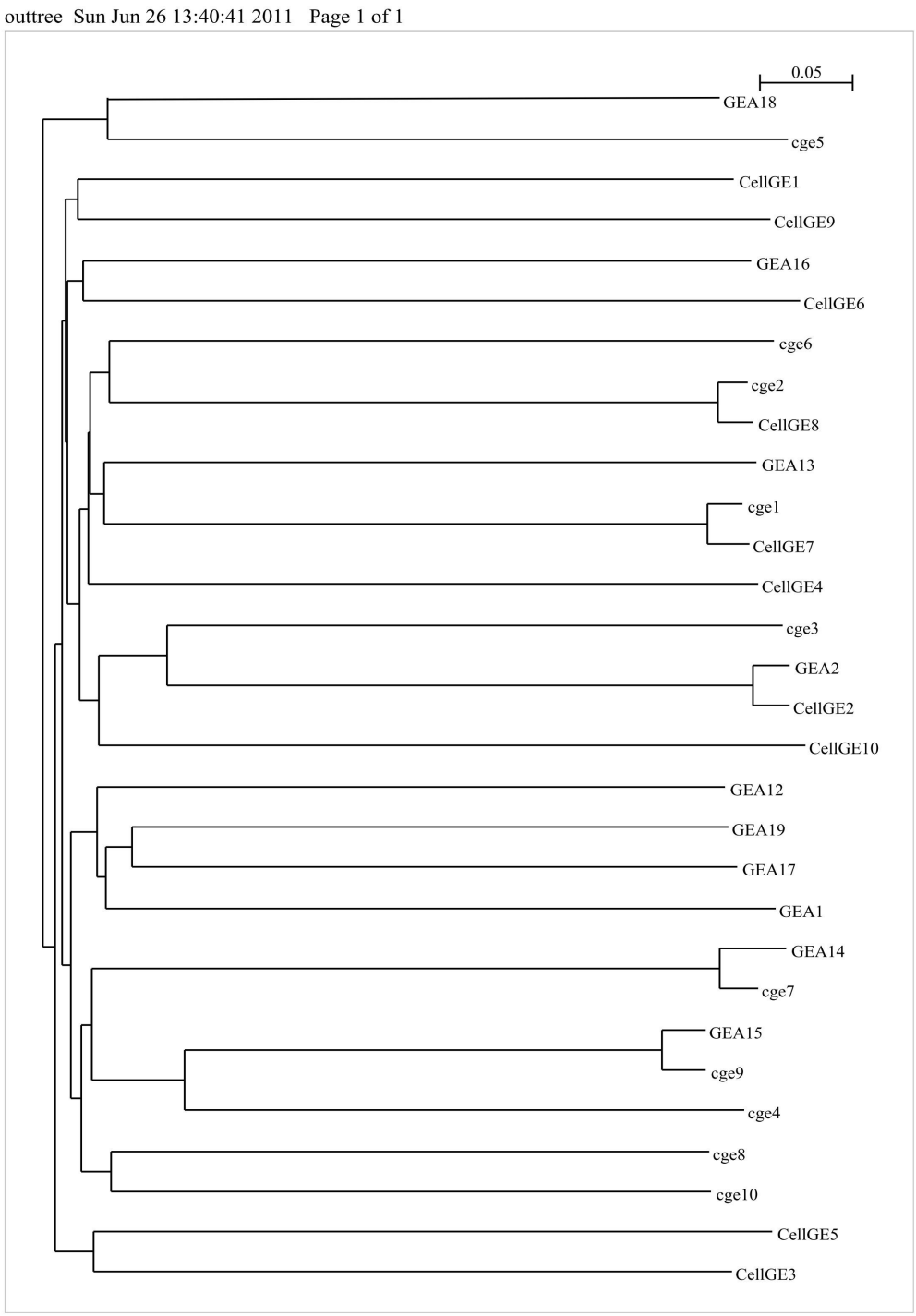


Abbildung 15: Baum mit einer Fehlerwahrscheinlichkeit von 0,2

Anhang 9: Baum mit Fehlerwahrscheinlichkeit 0,5

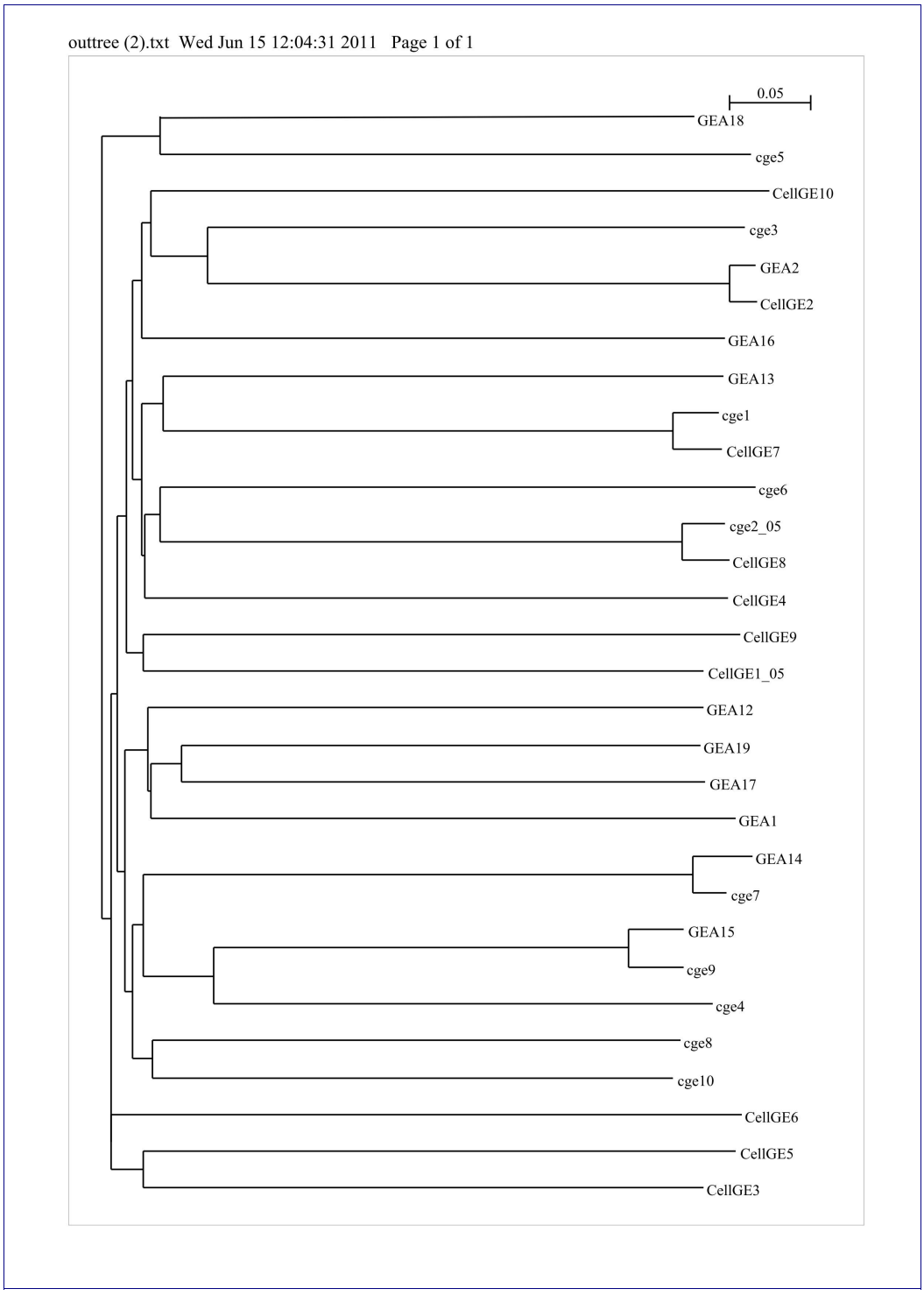


Abbildung 16: Baum mit einer Fehlerwahrscheinlichkeit von 0,5

Anhang 10: Baum mit Fehlerwahrscheinlichkeit 0,8

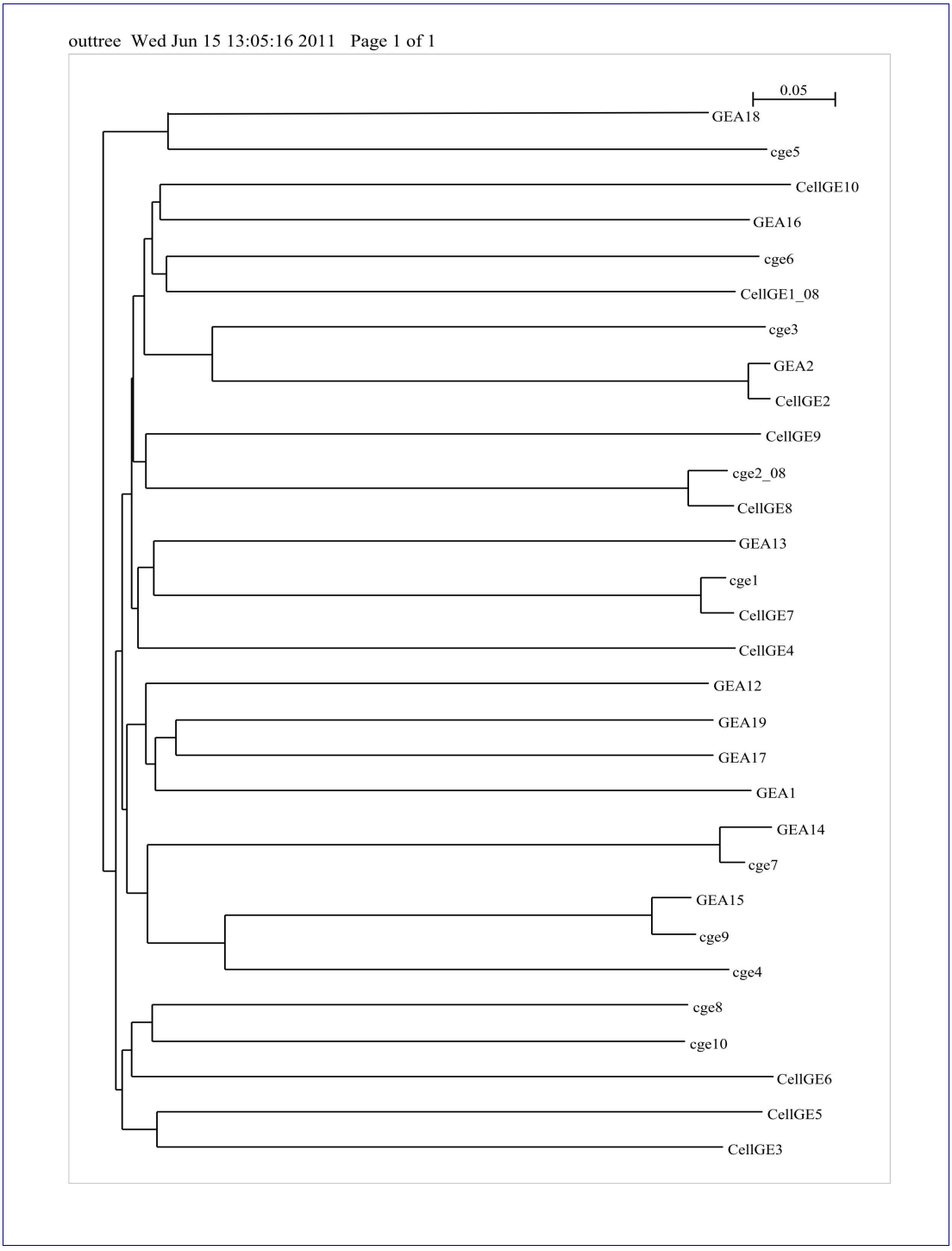


Abbildung 17: Baum mit einer Fehlerwahrscheinlichkeit von 0,8

Anhang 11: Ausgangsbaum mit optischer Hervorhebung

Abbildung 18: Ausgangsbaum mit farblicher Markierung

Anhang 12: Baum mit Fehlerwahrscheinlichkeit 0,5 und optischer Hervorhebung

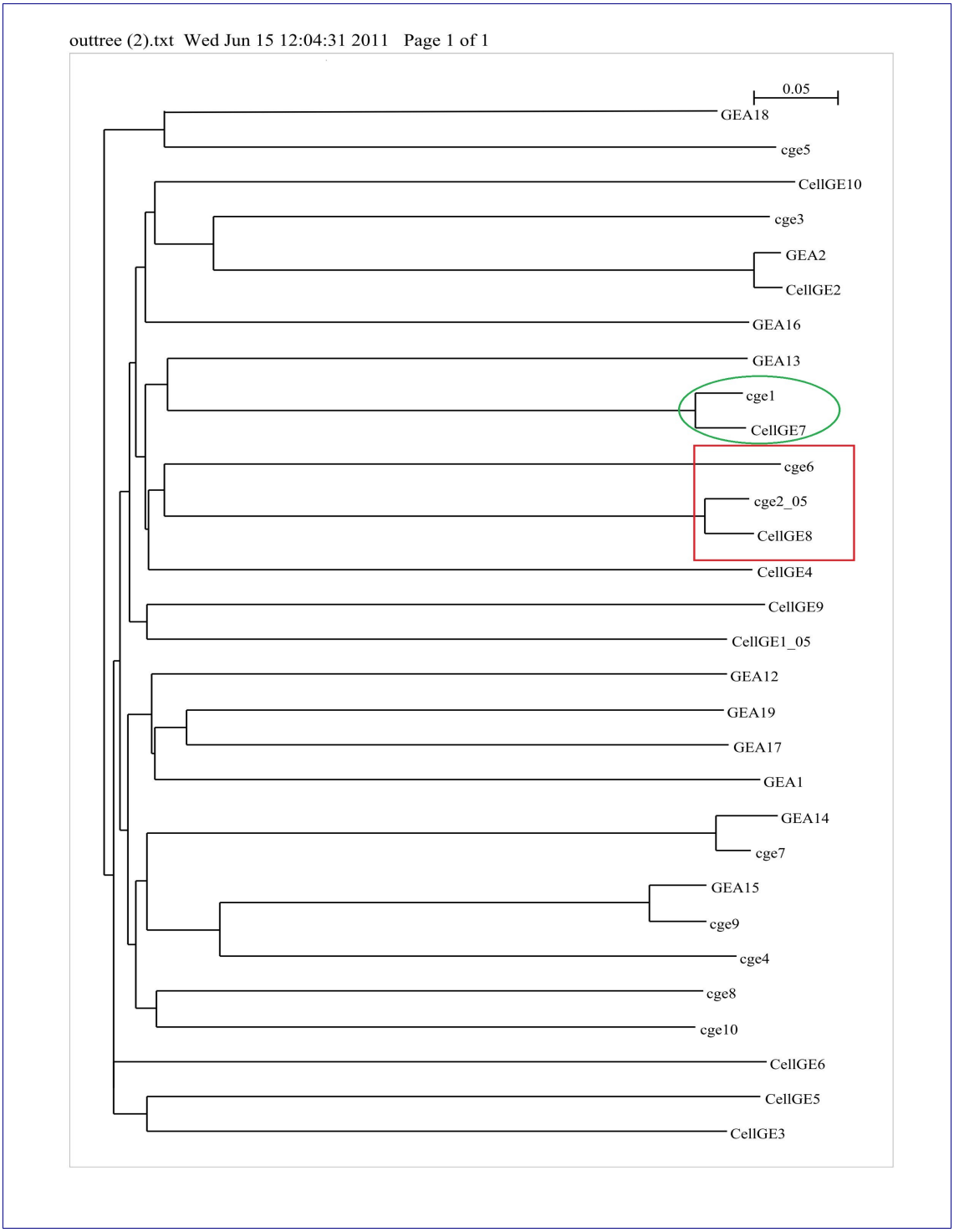


Abbildung 19: Baum mit einer Fehlerwahrscheinlichkeit von 0,5 und farblicher Markierung

Anhang 13: Baum mit verschiedenen Fehlerwahrscheinlichkeiten

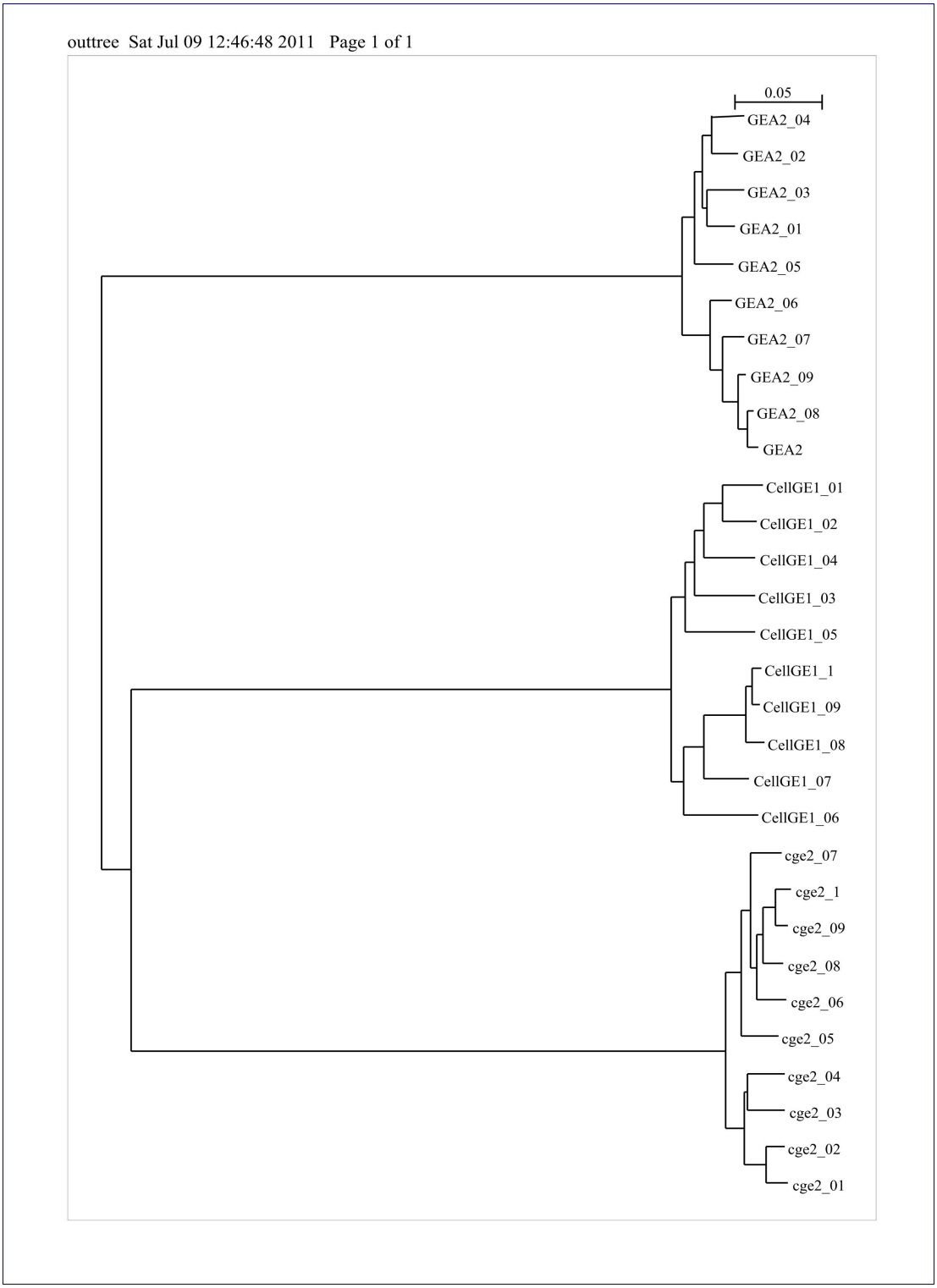


Abbildung 20: Baum mit verschiedenen Fehlerwahrscheinlichkeiten

Anhang 14: Baum mit zufälligen Fehlerwahrscheinlichkeiten

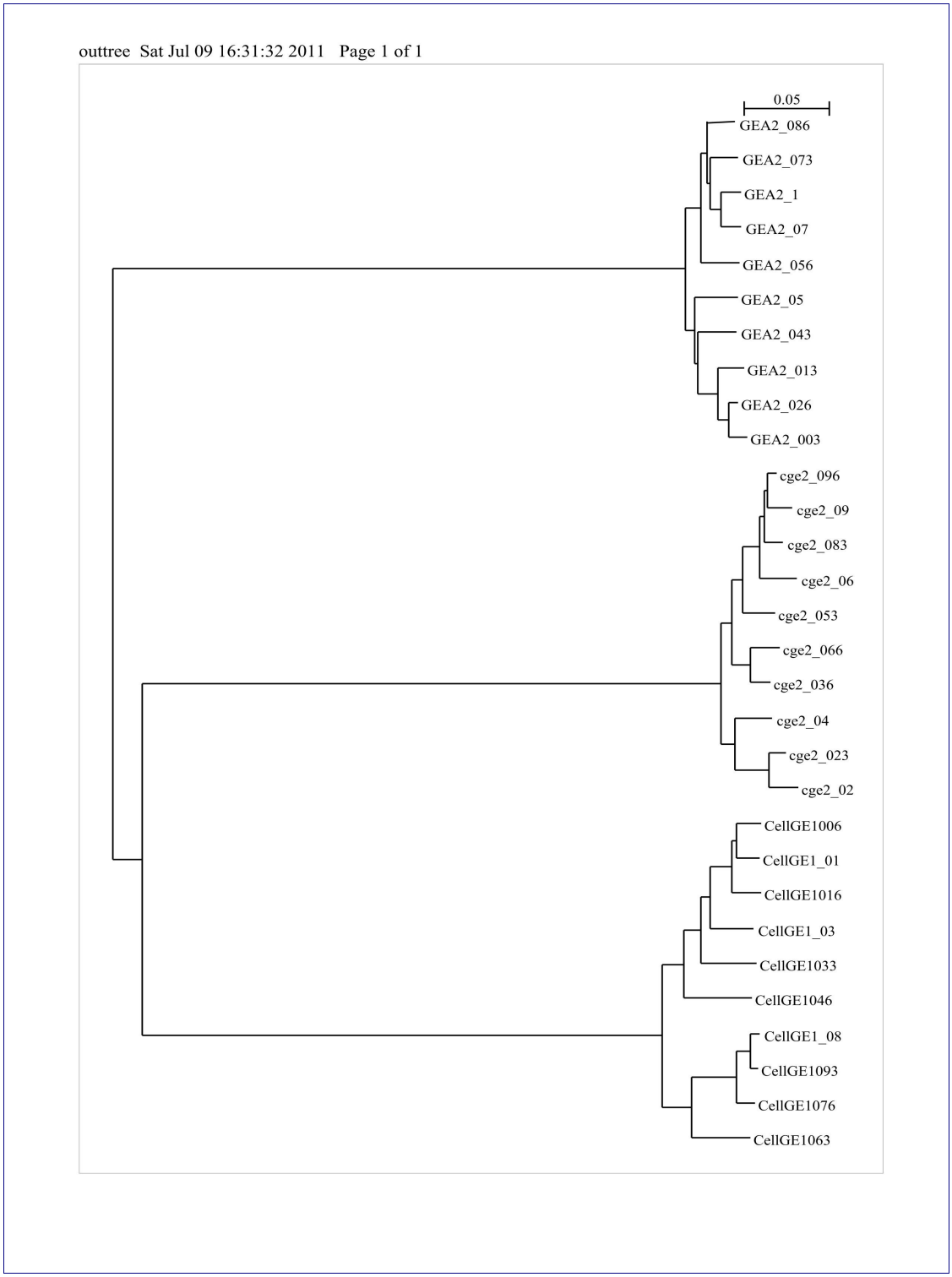


Abbildung 21: Baum mit zufälligen Fehlerwahrscheinlichkeiten

Anhang 15: Baum mit verschiedener Textverteilung und Fehlerwahrscheinlichkeit

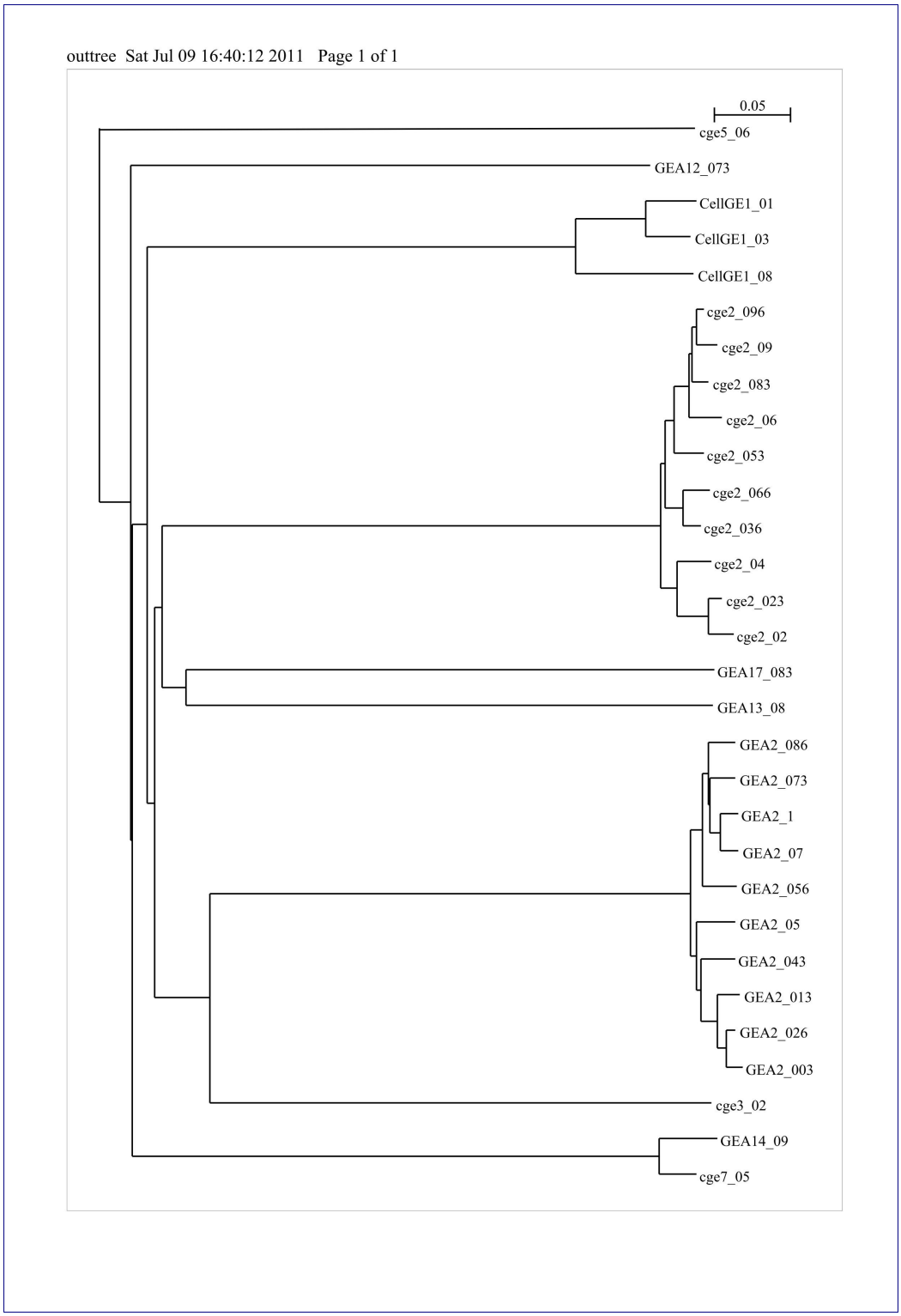


Abbildung 22: Baum mit verschiedenen Texthäufigkeiten und Fehlerwahrscheinlichkeiten

11 Literaturverzeichnis

[1] Heyer,Gerd: Quasthoff, Uwe: Wittig, Thomas: Text Mining: Wissensrohstoff Text.-1.Auflage.- Herdecke: W3L, 2006

[2] Verfasser unbekannt: Begriffsdefinition Ähnlichkeit, URL: <http://de.thefreedictionary.com/%C3%84hnlichkeit> verfügbar am 08.08.2011

[3] Haenelt, Karin: Ähnlichkeitsmaße für Vektoren.Kursfolien.2007

[4] Verfasser unbekannt: Tf-idf-weighting. URL: <http://nlp.stanford.edu/IRbook/html/htmledition/tf-idf-weighting-1.html> verfügbar am 10.05.2011

[5] Pedersen,Ted: README - General information about Text::Similarity. URL: <http://cpan.noris.de/authors/id/T/TP/TPEDERSE/Text-Similarity-0.08.readme> verfügbar am 10.05.2011

[6] Verfasser unbekannt: PHYLIP general information. URL: <http://evolution.genetics.washington.edu/phylip/general.html> verfügbar am 10.05.2011

[7] Opperdoes,Fred:Neighbor-joining-method. URL: <http://www.icp.ucl.ac.be/~opperd/private/neighbor.html> verfügbar am 10.05.2011

[8] Verfasser Unbekannt: Common Spelling Mistakes in IELTS Listening. URL:<http://easenglish.net/Files/IELTS/IELTS common Mistakes.pdf> verfügbar am 07.07.2011

Programmquellen:

[9] Ted Pedersen: Text Similarity 0.08. URL: <http://www.d.umn.edu/~tpederse/text-similarity.html> verfügbar am 29.05.2011

[10] Joseph Felsenstein: PHYLIP.

URL: <http://evolution.genetics.washington.edu/phylip/getme.html> verfügbar am 29.05.2011

[11] Perrière, G.: Gouy, M. (1996) WWW-Query: An on-line retrieval system for biological sequence banks. *Biochimie*, **78**, 364-369. URL:<http://pbil.univ-lyon1.fr/software/njplot.html> verfügbar am 29.05.2011